



(19)

(11) Publication number: **2001184327 A**

Generated Document.

## PATENT ABSTRACTS OF JAPAN

(21) Application number: **2000317826**

(51) Intl. Cl.: **G06F 15/177 G06F 9/445 G06F 12/00 G06F 15/16**

(22) Application date: **18.10.00**

(30) Priority: **18.10.9928.04.00 USUS  
1999 160104 2000 560199**

(43) Date of application  
publication: **06.07.01**

(84) Designated contracting  
states:

(71) Applicant: **FISHER ROSEMOUNT SYST INC**

(72) Inventor: **NIXON MARK  
CHATKOFF TERESA  
GILBERT STEVEN**

(74) Representative:

**(54) CONFIGURATION  
DATA BASE SYSTEM,  
DISTRIBUTED  
CONFIGURATION DATA  
BASE SYSTEM, METHOD  
FOR STORING AND USING  
CONFIGURATION DATA,  
DATA BASE SERVER,  
METHOD FOR PROVIDING  
ACCESS TO DATA BASE  
COMPONENT, METHOD  
FOR USING DATA BASE  
SERVER FOR PROVIDING  
PARALLEL ACCESS TO  
COMPONENT STORED IN  
DATA BASE AND PROCESS  
CONTROL SYSTEM**

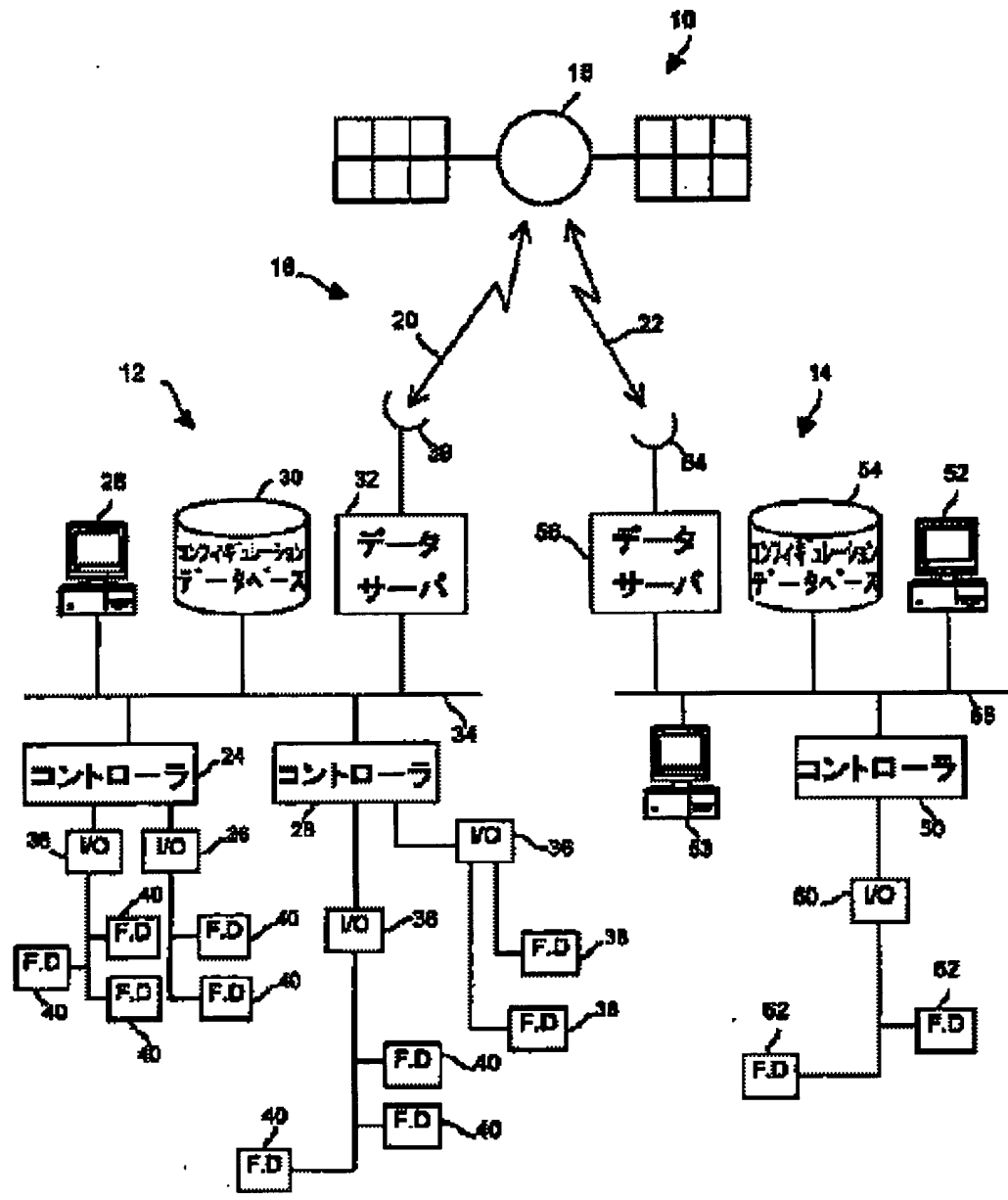
**(57) Abstract:**

**PROBLEM TO BE SOLVED:** To enable each of users at different positions to timely and safely change a configuration data base without conflict.

**SOLUTION:** This system is provided with plural data bases distributed at plural physical positions inside a process control system and each of data bases stores the subset of configuration data. This subset can be

accessed from the user on any arbitrary site inside the system. In order to prevent configuration data watched by a subscriber inside the system from getting old, the data base server automatically detects a change into each of items inside the data base and transmits the notice of the executed change to each of subscribers.

COPYRIGHT: (C)2001,JPO



(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開2001-184327

(P2001-184327A)

(43)公開日 平成13年7月6日(2001.7.6)

(51)Int.Cl. <sup>7</sup>	識別記号	F I	テーマコード(参考)
G 0 6 F 15/177	6 8 2	G 0 6 F 15/177	6 8 2 J
9/445		12/00	5 3 3 J
12/00	5 3 3	15/16	6 2 0 H
15/16	6 2 0	9/06	6 1 0 A

審査請求 未請求 請求項の数79 O L (全 47 頁)

(21)出願番号 特願2000-317826(P2000-317826)

(22)出願日 平成12年10月18日(2000.10.18)

(31)優先権主張番号 6 0 / 1 6 0 1 0 4

(32)優先日 平成11年10月18日(1999.10.18)

(33)優先権主張国 米国 (U S)

(31)優先権主張番号 0 9 / 5 6 0 1 9 9

(32)優先日 平成12年4月28日(2000.4.28)

(33)優先権主張国 米国 (U S)

(71)出願人 594120847

フィッシャー・ローズマウント システムズ, インコーポレイテッド

アメリカ合衆国 78754 テキサス オー  
スティン キャメロン ロード 8301

(72)発明者 ニクソン, マーク

アメリカ合衆国 78681 テキサス ラウ  
ンド ロック ブラックジャック ドライ  
ブ 1503

(74)代理人 100065868

弁理士 角田 嘉宏 (外4名)

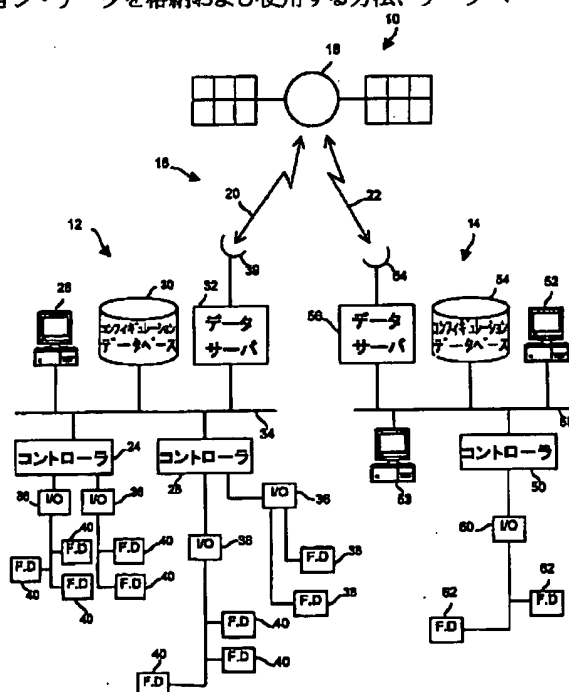
最終頁に続く

(54)【発明の名称】 コンフィギュレーション・データベース・システム、分散型のコンフィギュレーション・データベース・システム、コンフィギュレーション・データを格納および使用する方法、データベー

(57)【要約】 (修正有)

【課題】異なる位置の各々におけるユーザに、タイムリ  
ーで、安全で且つ非矛盾したコンフィギュレーションデー  
タベースの変更を可能とする。

【解決手段】プロセス制御システム内の複数の物理的な  
位置に分散された複数のデータベースを備え、各データ  
ベースは、コンフィギュレーション・データのサブセッ  
トを格納する。該サブセットには、前記システム内の任  
意のサイトでのユーザからアクセスすることができる。  
前記システム内で予約者によって見られているコンフィ  
ギュレーション・データが古くなることを防止すべく、  
前記データベース・サーバは、前記データベース内の各  
項目への変更を自動的に検出し、なされた変更の通知を  
予約者の各々に送信する。



## 【特許請求の範囲】

【請求項1】 複数の地理的に分散された物理的な位置を備えたプロセス制御システムに関連するコンフィギュレーション・データを格納すべく使用されるコンフィギュレーション・データベース・システムにおいて、前記複数の物理的な位置の各々に配置され、各々が前記コンフィギュレーション・データの異なる部分を元々格納するために構成されたコンフィギュレーション・データベースと、前記複数の地理的に分散された物理的な位置を相互に通信可能に結合する通信ネットワークと、該通信ネットワークを介して前記コンフィギュレーション・データベースの各々と通信し、コンフィギュレーション動作を行うべく、前記コンフィギュレーション・データベースのうちの2つ以上からのデータを使用すべく構成されたコンフィギュレーション・アプリケーションとを備えることを特徴とするコンフィギュレーション・データベース・システム。

【請求項2】 前記コンフィギュレーション・アプリケーションは、第1のコンフィギュレーション・データベースのうちの1つ内で前記コンフィギュレーション・データのサブセットを予約すべく構成され、前記第1のコンフィギュレーション・データベースのうちの1つは、前記コンフィギュレーション・アプリケーションが予約する該第1のコンフィギュレーション・データベースに格納された前記コンフィギュレーション・データのサブセットへの変更を自動的に認識する第1のルーチンと、前記コンフィギュレーション・アプリケーションに前記コンフィギュレーション・データのサブセットの変更を自動的に通知する第2のルーチンとを有するデータベース・サーバとを備えることを特徴とする請求項1記載のコンフィギュレーション・データベース・システム。

【請求項3】 前記第2のルーチンは、前記第1のルーチンが前記コンフィギュレーション・データのサブセットへの変更を検出した際、前記コンフィギュレーション・アプリケーションに、前記第1のコンフィギュレーション・データベースに格納された前記コンフィギュレーション・データのサブセットになされた変更を自動的に伝えることを特徴とする請求項2記載のコンフィギュレーション・データベース・システム。

【請求項4】 前記コンフィギュレーション・アプリケーションは、前記第1のコンフィギュレーション・データベース内のコンフィギュレーション・データの項目をロックするためのロック・コマンドを送信するロック部分を有し、前記データベース・サーバは、前記ロック・コマンドを送信した前記コンフィギュレーション・アプリケーション以外の如何なるコンフィギュレーション・アプリケーションによって前記コンフィギュレーション・データの項目が変更されることから防ぐために、前記第1のコンフィギュレーション・データベース内の前

記コンフィギュレーション・データの項目をロックするロック・ルーチンを更に有することを特徴とする請求項2記載のコンフィギュレーション・データベース・システム。

【請求項5】 第2のコンフィギュレーション・アプリケーションを更に備え、前記データベース・サーバは、第1および第2のコンフィギュレーション・アプリケーションに、前記第1のコンフィギュレーション・データベース内のコンフィギュレーション・データの特定の項目への並列アクセスを提供する第2のルーチンと、前記第1のコンフィギュレーション・アプリケーションが前記第1のコンフィギュレーション・データベース内のコンフィギュレーション・データの特定の項目に書き込むときに、前記コンフィギュレーション・データの特定の項目をロックし、それによって、前記第2のコンフィギュレーション・アプリケーションによって、前記コンフィギュレーション・データの特定の項目が変更されることを防止するロック・ルーチンとを有することを特徴とする請求項2記載のコンフィギュレーション・データベース・システム。

【請求項6】 前記コンフィギュレーション・データベースは、下位レベルにおいて少なくとも2つのコンフィギュレーション・データベースを有し、上位レベルにおいて少なくとも1つのコンフィギュレーション・データベースを有する階層を確立され、前記下位レベルにおけるコンフィギュレーション・データベースの各々が、前記上位レベルにおけるコンフィギュレーション・データベースに通信可能に結合されることを特徴とする請求項1記載のコンフィギュレーション・データベース・システム。

【請求項7】 前記下位レベルは、前記物理的な位置の異なるものに関連するゾーンを含み、第1のゾーンに関連するコンフィギュレーション・データは、該第1のゾーンでコンフィギュレーション・データベースに格納され、第1および第2のゾーンに関連するコンフィギュレーション・データは、該第1および第2のゾーンに通信可能に結合された階層における前記上位レベルでのコンフィギュレーション・データベースに格納されることを特徴とする請求項6記載のコンフィギュレーション・データベース・システム。

【請求項8】 前記下位レベルにおけるコンフィギュレーション・データベースと、前記上位レベルにおけるコンフィギュレーション・データベースとのうちの1つにおいて、同一の名前の使用を検出するソフトウェアを更に有していることを特徴とする請求項6記載のコンフィギュレーション・データベース・システム。

【請求項9】 前記コンフィギュレーション・アプリケーションは、前記コンフィギュレーション・データベースの第1のものを閲覧し、且つ前記コンフィギュレーション・データベースの第2のものを閲覧するように構成

されたブラウザを含むことを特徴とする請求項1記載のコンフィギュレーション・データベース・システム。

【請求項10】 前記コンフィギュレーション・データベースのうちの1つは、前記コンフィギュレーション・データベースの別の1つに元々格納された項目のローカル・コピーを格納することを特徴とする請求項1記載のコンフィギュレーション・データベース・システム。

【請求項11】 前記通信ネットワークは、前記コンフィギュレーション・データベースのうちの2つの間に、遅い通信回線を備えることを特徴とする請求項1のコンフィギュレーション・データベース・システム。

【請求項12】 複数の物理的な位置を有するプロセス制御システムで使用されるべく構成された分散型のコンフィギュレーション・データベース・システムにおいて、前記プロセス制御システムのためのコンフィギュレーション・データの第1の部分に格納する物理的な位置の第1のものに位置する第1のコンフィギュレーション・データベースと、

前記プロセス制御システムのためのコンフィギュレーション・データの第2の部分とは異なる第2の部分に格納する物理的な位置の第2のものに位置する第2のコンフィギュレーション・データベースと、遅い通信回線を使用して第1の物理的な位置と第2の物理的な位置とを相互接続する通信ネットワークとを備え、前記第1および第2のコンフィギュレーション・データベースは、前記第1および第2の物理的な位置でのユーザと通信すべく構成されることを特徴とする分散型のコンフィギュレーション・データベース・システム。

【請求項13】 前記第2のコンフィギュレーション・データベースは、前記第1のコンフィギュレーション・データベース内のコンフィギュレーション・データを予約すべく構成され、前記第1のコンフィギュレーション・データベースは、前記第2のコンフィギュレーション・データベースが予約する前記第1のコンフィギュレーション・データベースに格納された前記コンフィギュレーション・データへの変更を自動的に検出する第1のルーチンと、前記第2のコンフィギュレーション・データベースが予約する前記第1のコンフィギュレーション・データベースに格納された前記コンフィギュレーション・データへの変更を前記第2のコンフィギュレーション・データベースに自動的に通知する第2のルーチンとを有するデータベース・サーバを備えることを特徴とする請求項12記載の分散型のコンフィギュレーション・データベース・システム。

【請求項14】 前記第1のルーチンが、前記第2のコンフィギュレーション・データベースが予約する前記第1のコンフィギュレーション・データベースに格納された前記コンフィギュレーション・データへの変更を検出したときに、前記第2のルーチンは、前記第1のコンフィギ

ュレーション・データベースに格納された前記コンフィギュレーション・データになされた変更を前記第2のコンフィギュレーション・データベースへ自動的に伝えることを特徴とする請求項13記載の分散型のコンフィギュレーション・データベース・システム。

【請求項15】 前記第1のコンフィギュレーション・データベースに関連付けられた第1および第2のコンフィギュレーション・アプリケーション並びにデータベース・サーバを更に備え、該データベース・サーバは、前記第1および第2のコンフィギュレーション・アプリケーションに、前記第1のデータベース内のコンフィギュレーション・データの任意の特定の項目への並列アクセスを提供する共有キャッシュと、前記第1のコンフィギュレーション・アプリケーションがコンフィギュレーション・データの第1の特定の項目に書き込むときに、前記第1のコンフィギュレーション・データベース内のコンフィギュレーション・データの第1の特定の項目をロックすることによって、前記第2のコンフィギュレーション・アプリケーションがコンフィギュレーション・データの第1の特定の項目を変更することを防止するロック・ルーチンとを備えることを特徴とする請求項12記載の分散型のコンフィギュレーション・データベース・システム。

【請求項16】 第3の物理的な位置に配置された第3のコンフィギュレーション・データベースを更に備え、前記第1、第2、および第3のコンフィギュレーション・データベースは、前記第1および第2のコンフィギュレーション・データベースが下位レベルにあり、前記第3のコンフィギュレーション・データベースが上位レベルにある階層に構築され、前記下位レベルにおける第1および第2のコンフィギュレーション・データベースの各々が、通信ネットワークを介して、前記第3のコンフィギュレーション・データベースに前記上位レベル内で通信可能に結合されることを特徴とする請求項12記載の分散型のコンフィギュレーション・データベース・システム。

【請求項17】 前記下位レベルは、第1および第2のゾーンを含み、該第1のゾーンに関連するコンフィギュレーション・データは、前記第1のコンフィギュレーション・データベースに格納され、前記第2のゾーンに関連するコンフィギュレーション・データは、前記第2のコンフィギュレーション・データベースに格納され、前記第1および第2のゾーンに関連するコンフィギュレーション・データが、前記第3のコンフィギュレーション・データベースに格納されることを特徴とする請求項16記載の分散型のコンフィギュレーション・データベース・システム。

【請求項18】 前記第1および第3のコンフィギュレーション・データベース内で、又は前記第2および第3のコンフィギュレーション・データベース内で、同一の名前

の使用を検出するルーチンを更に備えることを特徴とする請求項16記載の分散型のコンフィギュレーション・データベース・システム。

【請求項19】 プロセス制御システムが、地理的に分離した2つ以上の物理的な位置を有する場合、前記プロセス制御システムに関連するコンフィギュレーション・データを格納および使用する方法において、複数のコンフィギュレーション・データベースの各々に前記コンフィギュレーション・データの異なる部分を格納し、前記複数のコンフィギュレーション・データベースのうちの2つが、異なる物理的な位置に配置されるステップと、前記複数のコンフィギュレーション・データベースの各々の間の通信回線を提供するステップと、コンフィギュレーション動作を行うために、前記複数のコンフィギュレーション・データベースのうちの2つ以上から、異なるコンフィギュレーション・データに同時にアクセスするステップとを有することを特徴とする方法。

【請求項20】 前記異なるコンフィギュレーション・データに同時にアクセスするステップは、アクセスされたコンフィギュレーション・データに前記複数のコンフィギュレーション・データベースのうちの2つ以上から予約するアプリケーションを使用し、前記アクセスされたコンフィギュレーション・データになされた任意の変更を前記複数のコンフィギュレーション・データベースのうちの2つ以上から前記アプリケーションに自動的に送信することを特徴とする請求項19記載の方法。

【請求項21】 前記コンフィギュレーション・データベース内に、下位レベルに2つのコンフィギュレーション・データベースを含み、上位レベルに1つのコンフィギュレーション・データベースを含む階層を確立するステップと、前記下位レベルの2つのコンフィギュレーション・データベースの各々と前記上位レベルの1つのコンフィギュレーション・データベースとの間の通信を構築するステップとを更に有することを特徴とする請求項19記載の方法。

【請求項22】 前記階層の上位レベルでの前記1つのコンフィギュレーション・データベース内で、および前記階層の下位レベルでの前記2つのコンフィギュレーション・データベース内で、コンフィギュレーション・データのためのユニークな名前を使用するステップを更に含み、同一の名前が、前記階層の下位レベルでの前記2つのコンフィギュレーション・データベースにおいて使用されることを特徴とする請求項21記載の方法。

【請求項23】 複数のクライアントに、データベースに格納されたデータベース・コンポーネントへの並列ア

クセスを提供すべく構成されたデータベース・サーバにおいて、

前記データベース・コンポーネントのコピーを保持する、前記データベースに通信可能に結合されたメモリ・ストアを含む共有キャッシュと、

前記メモリ・ストアに関連付けられ、該メモリ・ストアへのアクセスを制御するメモリ・ストア・ゲートと、各々が、前記メモリ・ストアへのアクセスを得るために前記メモリ・ストア・ゲートと通信し、前記メモリ・ストアに格納されたデータベース・コンポーネントに関して前記複数のクライアントのうちの1つと通信するサーバ・コンポーネントを含む、前記複数のクライアントのためのクライアント・スレッドとを備え、

前記複数のクライアントのうちの1つだけが前記データベース・コンポーネントにアクセスしているときに、前記メモリ・ストア・ゲートが単一のサーバ・コンポーネントと通信し、2つ以上のクライアントが前記データベース・コンポーネントにアクセスしているときに、前記メモリ・ストア・ゲートが2つ以上のサーバ・コンポーネントと通信することを特徴とするデータベース・サーバ。

【請求項24】 前記データベース内のデータベース・コンポーネントになされた変更を検出し、検出された変更を前記メモリ・ストア・ゲートに通知する通知エンジンを有する通知スレッドを更に備えることを特徴とする請求項23記載のデータベース・サーバ。

【請求項25】 前記メモリ・ストア・ゲートは、前記検出された変更について、前記メモリ・ストア・ゲートと通信するサーバ・コンポーネントに通知し、通知されたサーバ・コンポーネントは、該サーバ・コンポーネントが通信するクライアントに前記検出された変更について通知することを特徴とする請求項24記載のデータベース・サーバ。

【請求項26】 前記通知スレッドは、前記データベースに格納された前記データベース・コンポーネントを前記メモリ・ストア・ゲートにマッピングするマップを更に含み、前記通知エンジンは、前記検出された変更について前記メモリ・ストア・ゲートに通知するために前記マップを使用することを特徴とする請求項24記載のデータベース・サーバ。

【請求項27】 前記メモリ・ストア・ゲートは、ユニークな識別子を含むとともに、該ユニークな識別子を前記マップに登録し、前記メモリ・ストアは、前記データベースから前記データベース・コンポーネントを読み出すために、又は前記データベースへ前記データベース・コンポーネントを書き込むために、処理オブジェクトを使用し、該処理オブジェクトは、前記ユニークな識別子に関連付けられた前記マップ内の前記データベース・コンポーネントのためのデータベース・コンポーネント位置に登録することを特徴とする請求項26記載のデータ

ベース・サーバ。

【請求項28】 前記データベースの外で実行されるアプリケーションによって作られたデータベース・コンポーネントに関してなされた変更を検出し、前記データベース・コンポーネントでの検出された変更について前記メモリ・ストア・ゲートに通知するランタイム通知エンジンを含むランタイム・スレッドを更に有することを特徴とする請求項24記載のデータベース・サーバ。

【請求項29】 前記クライアント・スレッドの各々のサーバ・コンポーネントは、前記メモリ・ストアから読み出すこと、および書き込むことができることを特徴とする請求項23記載のデータベース・サーバ。

【請求項30】 前記サーバ・コンポーネントのうちの1つは、遅い通信回線を介して前記複数のクライアントのうちの1つと通信することを特徴とする請求項23記載のデータベース・サーバ。

【請求項31】 前記遅い通信回線は、衛星通信回線であることを特徴とする請求項23記載のデータベース・サーバ。

【請求項32】 前記メモリ・ストア・ゲートが前記メモリ・ストアに書き込むことを防止させるロック・マネージャを更に備えることを特徴とする請求項23記載のデータベース・サーバ。

【請求項33】 前記ロック・マネージャは、前記メモリ・ストア・ゲートに、前記複数のクライアントのうちの1つが前記メモリ・ストアに書き込んでいるときに、前記複数のクライアントのうちの1つを除く前記複数のクライアントの各々によって前記メモリ・ストアに書き込むことを防止させることを特徴とする請求項32記載のデータベース・サーバ。

【請求項34】 前記データベースは、コンフィギュレーション・データベースであり、前記データベース・コンポーネントは、プロセス制御ネットワークのコンフィギュレーションに関連するコンフィギュレーション・コンポーネントであることを特徴とする請求項23記載のデータベース・サーバ。

【請求項35】 複数のクライアントに、データベースに格納された複数のデータベース・コンポーネントへの並列アクセスを提供すべく構成されたデータベース・サーバにおいて、各々が前記複数のデータベース・コンポーネントの異なる1つのコピーを保持する、前記データベースに通信可能に結合された複数のメモリ・ストアを含む共有キャッシュと、前記複数のメモリ・ストアの各々に関連付けられ、各々が、関連付けられたメモリ・ストアへのアクセスを制御するメモリ・ストア・ゲートと、各々が、前記メモリ・ストア・ゲートのうちの1つに関連付けられたメモリ・ストアへのアクセスを得るために前記メモリ・ストア・ゲートのうちの1つと通信し、前

記メモリ・ストア・ゲートのうちの1つに関連付けられた前記メモリ・ストアに格納されたデータベース・コンポーネントに関して前記複数のクライアントのうちの1つと通信するサーバ・コンポーネントを含む、複数のクライアントの各々のためのクライアント・スレッドとを備え、

前記複数のクライアントのうちの1つだけが前記メモリ・ストア・ゲートに関連付けられたメモリ・ストアに格納されたデータベース・コンポーネントにアクセスしているときに、前記メモリ・ストア・ゲートの各々が単一のサーバ・コンポーネントと通信し、前記複数のクライアントのうちの2つ以上が前記メモリ・ストア・ゲートに関連付けられたメモリ・ストアに格納されたデータベース・コンポーネントにアクセスしているときに、前記メモリ・ストア・ゲートの各々が2つ以上のサーバ・コンポーネントと通信することを特徴とするデータベース・サーバ。

【請求項36】 前記複数のデータベース・コンポーネントのうちの1つになされた変更を検出し、検出された変更を前記メモリ・ストア・ゲートのうちの1つに通知する通知エンジンを有する通知スレッドを更に備えることを特徴とする請求項35記載のデータベース・サーバ。

【請求項37】 前記メモリ・ストア・ゲートのうち1つは、該メモリ・ストア・ゲートのうちの1つと通信する前記サーバ・コンポーネントに前記検出された変更について通知し、通知されたサーバ・コンポーネントは、該サーバ・コンポーネントが通信するクライアントに前記検出された変更について通知することを特徴とする請求項36記載のデータベース・サーバ。

【請求項38】 前記通知されたサーバ・コンポーネントは、該サーバ・コンポーネントが通信するクライアントに、前記検出された変更を提供することを特徴とする請求項37記載のデータベース・サーバ。

【請求項39】 前記メモリ・ストア・ゲートの各々が前記関連付けられたメモリ・ストアに書き込むことを防止するロック・マネージャを更に備えることを特徴とする請求項35記載のデータベース・サーバ。

【請求項40】 一方が親メモリ・ストア・ゲートであり、他方が子メモリ・ストア・ゲートであるように、前記メモリ・ストア・ゲートのうちの2つが親子関係を有し、前記ロック・マネージャが前記親メモリ・ストア・ゲートをロックさせるときに、前記ロック・マネージャが前記子メモリ・ストア・ゲートをロックさせることを特徴とする請求項39記載のデータベース・サーバ。

【請求項41】 複数のクライアントに、データベースに格納されたデータベース・コンポーネントへの並列アクセスを提供するために、プロセス制御システムにおいて使用されるべく構成されたデータベース・サーバにおいて、

共有キャッシュと、

前記データベース・コンポーネントが複数のクライアントのうちの1つ以上によってアクセスされているときに、前記共有キャッシュ内で、前記データベース・コンポーネントのためのメモリ・ストアを確立する第1の要素と、

前記複数のクライアントの各々のためのクライアント・スレッドを確立するクライアント・スレッド・ルーチンとを備え、

前記クライアント・スレッドの各々は、前記クライアントのうちの1つと通信するサーバ・コンポーネントを含み、

前記クライアント・スレッド・ルーチンは、前記メモリ・ストアと通信し、これによって、前記複数のクライアントのうちの1つによって前記メモリ・ストアに格納されたデータベース・コンポーネントへのアクセスを提供し、

前記複数のクライアントのうちの1つだけがデータベース・コンポーネントにアクセスしているときに、単一のサーバ・コンポーネントと通信すべく構成され、前記複数のクライアントのうちの2つ以上がデータベース・コンポーネントにアクセスしているときに、前記メモリ・ストアが2つ以上のサーバ・コンポーネントと通信すべく構成されていることを特徴とするデータベース・サーバ。

【請求項42】 前記第1の要素は、前記メモリ・ストアに関連付けられたメモリ・ストア・ゲートを確立し、該メモリ・ストア・ゲートは、2つ以上のクライアントが前記データベース・コンポーネントにアクセスしているときに、前記2つ以上のクライアント・スレッドにおける前記サーバ・コンポーネントの各々によって前記関連付けられたメモリ・ストアへのアクセスを提供するために、前記クライアント・スレッドのうちの2つ以上における前記サーバ・コンポーネントと通信すべく構成されることを特徴とする請求項41記載のデータベース・サーバ。

【請求項43】 前記メモリ・ストアは、前記データベース内のデータベース・コンポーネントにアクセスするための処理オブジェクトを使用することを特徴とする請求項42記載のデータベース・サーバ。

【請求項44】 通知エンジンおよびマップを有する通知スレッドを更に備え、前記通知エンジンは、前記データベース内のデータベース・コンポーネントへの変更を検出し、検出された変更について前記メモリ・ストア・ゲートに通知すべくマップを使用することを特徴とする請求項42記載のデータベース・サーバ。

【請求項45】 前記メモリ・ストア・ゲートは、該メモリ・ストア・ゲートが通信する前記サーバ・コンポーネントの全てに前記データベース・コンポーネントの前記検出された変更について通知し、通知されたサーバ・

コンポーネントは、該サーバ・コンポーネントが通信するクライアントに前記検出された変更について通知することを特徴とする請求項44記載のデータベース・サーバ。

【請求項46】 前記通知エンジンが前記データベース内のデータベース・コンポーネントの前記検出された変更について前記メモリ・ストア・ゲートに通知するときに、該メモリ・ストア・ゲートが前記メモリ・ストアに前記データベースから前記データベース・コンポーネントを読み込ませることを特徴とする請求項44記載のデータベース・サーバ。

【請求項47】 前記通知スレッドは、前記データベース内のデータベース・コンポーネントへの変更を検出するために、前記データベースによって生成された変更リストを使用することを特徴とする請求項44記載のデータベース・サーバ。

【請求項48】 前記プロセス制御システム内で実行されるランタイム・アプリケーションの操作に基づいて前記データベース・コンポーネントへの変更を検出し、検出された変更について前記メモリ・ストア・ゲートに通知するランタイム・サービス通知スレッドを更に含むことを特徴とする請求項44記載のデータベース・サーバ。

【請求項49】 前記サーバ・コンポーネントのうちの少なくとも1つは、衛星通信回線を介して前記複数のクライアントのうちの1つと通信すべく構成されることを特徴とする請求項41記載のデータベース・サーバ。

【請求項50】 前記サーバ・コンポーネントのうちの少なくとも1つは、セルラー通信回線を介して前記複数のクライアントのうちの1つと通信すべく構成されることを特徴とする請求項41記載のデータベース・サーバ。

【請求項51】 前記サーバ・コンポーネントのうちの少なくとも1つは、無線通信回線を介して前記複数のクライアントのうちの1つと通信すべく構成されることを特徴とする請求項41記載のデータベース・サーバ。

【請求項52】 前記サーバ・コンポーネントのうちの少なくとも1つは、電話通信回線を介して前記複数のクライアントのうちの1つと通信すべく構成されることを特徴とする請求項41記載のデータベース・サーバ。

【請求項53】 前記複数のクライアントのうちの1つが前記複数のクライアントのうちの別のものによる前記メモリ・ストアへのアクセスを防止すべく前記メモリ・ストアにアクセスしているときに、前記メモリ・ストアをロックするロック・マネージャを更に備えることを特徴とする請求項41記載のデータベース・サーバ。

【請求項54】 コンテキスト・メモリを更に備え、前記メモリ・ストアが、前記クライアントのうちの第1のものに代わって前記データベースと通信するときに、前記コンテキスト・メモリの第1の部分を使用し、前記ク



クライアントのうちの第2のものに代わって前記コンフィギュレーション・データベースと通信するときに、前記コンテキスト・メモリの前記第1の部分とは異なる第2の部分を使用することを特徴とする請求項4記載のデータベース・サーバ。

【請求項5】 複数のクライアントに、データベースに格納されたデータベース・コンポーネントへのアクセスを提供する方法において、  
前記データベース・コンポーネントに関して前記データベースからの読み出し又は前記データベースへの書き込みを行うべく、共有キャッシュ内に前記データベースと通信する格納オブジェクトを生成するステップと、  
前記複数のクライアントの第1のものと前記データベース・コンポーネントとの間の通信を提供するために第1のクライアント・スレッドを使用するステップと、  
前記複数のクライアントの第2のものと前記データベース・コンポーネントとの間の通信を提供するために第2のクライアント・スレッドを使用するステップとを有し、  
前記第1のクライアント・スレッドを使用するステップは、  
前記データベース・コンポーネントに関して前記第1のクライアントと通信する第1の通信コンポーネントを生成するステップと、  
該第1の通信コンポーネントと前記格納オブジェクトとの間のリンクを提供するステップとを有し、  
前記第2のクライアント・スレッドを使用するステップは、  
前記データベース・コンポーネントに関して前記第2のクライアントと通信する第2の通信コンポーネントを生成するステップと、  
該第2の通信コンポーネントと前記格納オブジェクトとの間のリンクを提供するステップとを有することを特徴とする方法。

【請求項6】 前記格納オブジェクトを生成するステップは、前記格納オブジェクトに関連付けられた格納オブジェクト・ゲートを生成するステップを含み、該格納オブジェクト・ゲートは、前記第1および第2のクライアント・スレッドにおける前記サーバ・コンポーネントの各々によって前記格納オブジェクトへのアクセスを提供するために、前記第1および第2のクライアント・スレッドにおける前記サーバ・コンポーネントと通信すべく構成されることを特徴とする請求項5記載の方法。

【請求項7】 前記データベース内のデータベース・コンポーネントにアクセスし、前記格納オブジェクトと前記データベースとの間の通信を提供するための処理オブジェクトを使用するステップを更に有することを特徴とする請求項5記載の方法。

【請求項8】 前記データベース内のデータベース・コンポーネントへの変更を検出し、前記格納オブジェク

ト・ゲートに前記変更について通知するステップを更に有することを特徴とする請求項5記載の方法。

【請求項9】 前記格納オブジェクト・ゲートが通信する前記サーバ・コンポーネントの全てに、前記データベース・コンポーネントで検出された変更について通知する前記格納オブジェクト・ゲートを使用するステップと、通知されたサーバ・コンポーネントから該通知されたサーバ・コンポーネントが通信するクライアントへ、前記データベース・コンポーネントに検出された変更を伝えるステップとを更に有することを特徴とする請求項5記載の方法。

【請求項10】 前記データベース・コンポーネントへの変更が検出されたときに、前記格納オブジェクトに、前記データベースから前記データベース・コンポーネントを読み出させるステップを更に有することを特徴とする請求項5記載の方法。

【請求項11】 前記通知されたサーバ・コンポーネントから前記クライアントへ前記検出された変更を伝えるステップは、前記サーバ・コンポーネントのうちの1つと前記クライアントのうちの1つとの間で通信するために衛星通信回線を使用するステップを含むことを特徴とする請求項5記載の方法。

【請求項12】 前記通知されたサーバ・コンポーネントから前記クライアントへ前記検出された変更を伝えるステップは、前記サーバ・コンポーネントのうちの1つと前記クライアントのうちの1つとの間で通信するために無線通信回線を使用するステップを有することを特徴とする請求項5記載の方法。

【請求項13】 前記通知されたサーバ・コンポーネントから前記クライアントへ前記検出された変更を伝えるステップは、前記サーバ・コンポーネントのうちの1つと前記クライアントのうちの1つとの間で通信するために電話回線を使用するステップを更に有することを特徴とする請求項5記載の方法。

【請求項14】 前記複数のクライアントのうちの1つが前記複数のクライアントのその他ものによる前記メモリ・オブジェクトへのアクセスを防止するために前記格納オブジェクトにアクセスしているときに、前記格納オブジェクトをロックするステップを更に有することを特徴とする請求項5記載の方法。

【請求項15】 複数のクライアントに、共有キャッシュを有するデータベースに格納されたコンポーネントへの並列アクセスを提供するためにデータベース・サーバを使用する方法において、  
クライアントが前記データベース内のコンポーネントへのアクセスを要求するときに、

(1) 前記コンポーネントに関してクライアントと通信するデータベース・サーバ内に通信コンポーネント・オブジェクトを生成し、

(2) 共有メモリ・オブジェクトが前記共有キャッシュ内

で確立されなかったか否かを判断し、前記共有メモリ・オブジェクトが、前記共有キャッシュ内で確立されなかった場合に、前記共有キャッシュ内の前記コンポーネントのために前記共有メモリを生成し、

(3) 前記共有メモリ・オブジェクトと前記通信コンポーネントと間に通信回線を提供するステップと、前記データベースから読み出し、該データベースへ書き込むための前記共有メモリ・オブジェクトを使用し、これによって、前記データベースから前記コンポーネントを読み出し、前記データベース内へ前記コンポーネントを書き込むステップとを有し、

2つ以上のクライアントが同一のコンポーネントにアクセスするときに、2つ以上の通信コンポーネント・オブジェクトは、同一の共有メモリ・オブジェクトに通信可能に結合されることを特徴とする方法。

【請求項66】 前記コンポーネントのうちの1つへの変更を検出し、前記コンポーネントのうちの1つにアクセスする前記クライアントの各々に、前記検出された変更について通知するステップを更に有することを特徴とする請求項65記載の方法。

【請求項67】 前記コンポーネントのうちの1つのための共有メモリ・オブジェクトに、前記データベースから前記変更されたコンポーネントを読み出させるステップを更に有することを特徴とする請求項66記載の方法。

【請求項68】 前記通信コンポーネントのうちの1つが前記共有メモリ・オブジェクトのうちの1つにアクセスするときに、前記共有メモリ・オブジェクトのうちの1つをロックし、これによって、前記通信オブジェクトのうちの1つが前記共有メモリ・オブジェクトにアクセスするときに、前記通信オブジェクトのその他のものが前記共有メモリ・コンポーネントにアクセスすることを防止するステップを更に有することを特徴とする請求項66記載の方法。

【請求項69】 第1の物理的な位置に配置され、データベース・コンポーネントを格納するデータベースと、そのうちの1つが前記第1の物理的な位置から実質的に地理的に隔離した第2の物理的な位置に配置された複数のクライアント・アプリケーションと、

前記第1の物理的な位置および前記第2の物理的な位置の通信回線と、

前記複数のクライアント・アプリケーションによる前記データベース内の前記データベース・コンポーネントへのアクセスを提供するデータベース・サーバとを備え、該データベース・サーバは、

複数のメモリ・オブジェクトを有し、該複数のメモリ・オブジェクトの各々がデータベースと通信して、前記複数のクライアントのうちの少なくとも1つによってアクセスされているデータベース・コンポーネントのうちの1つのコピーを格納する共有キャッシュと、

前記クライアント・アプリケーションの各々に関連付けられ、各々が、対応するクライアント・アプリケーションのうちの1つおよび前記メモリ・オブジェクトのうちの1つと通信する一又は複数の通信オブジェクトとを備え、2つ以上のクライアント・アプリケーションによってアクセスされているメモリ・オブジェクトが、2つ以上の通信オブジェクトに通信可能に結合されていることを特徴とするプロセス制御システム。

【請求項70】 前記データベース・サーバは、前記データベース・コンポーネントのうちの1つへの変更を検出し、前記データベース・コンポーネントのうちの1つへの変更の存在を、該データベース・コンポーネントのうちの1つにアクセスしているクライアントの各々に自動的に通知する通知ルーチンを更に備えることを特徴とする請求項69記載のプロセス制御システム。

【請求項71】 前記通知ルーチンは、前記変更の後、前記データベース・コンポーネントのうちの1つの状態を、該データベース・コンポーネントのうちの1つにアクセスしているクライアントの各々に通知することを特徴とする請求項70記載のプロセス制御システム。

【請求項72】 前記データベースは、コンフィギュレーション・データベースであり、前記データベース・コンポーネントは、コンフィギュレーション・コンポーネントであることを特徴とする請求項70記載のプロセス制御システム。

【請求項73】 前記通信オブジェクトの各々は、対応するメモリ・オブジェクトのうちの1つに読み書きすべく構成されることを特徴とする請求項70記載のプロセス制御システム。

【請求項74】 前記データベース・サーバは、第2の通信コンポーネントのうちの1つがメモリ・オブジェクトのうちの1つにアクセスしているときに、第1の通信コンポーネントのうちの1つが前記メモリ・オブジェクトのうちの1つにアクセスすることを防ぐロック・マネージャを更に備えることを特徴とする請求項70記載のプロセス制御システム。

【請求項75】 前記通信回線は、衛星通信回線であることを特徴とする請求項69記載のプロセス制御システム。

【請求項76】 前記通信回線は、セルラー通信回線であることを特徴とする請求項69記載のプロセス制御システム。

【請求項77】 前記通信回線は、電話回線であることを特徴とする請求項69記載のプロセス制御システム。

【請求項78】 前記通信回線は、無線通信回線であることを特徴とする請求項69記載のプロセス制御システム。

【請求項79】 前記通信回線は、広域ネットワーク回線であることを特徴とする請求項69記載のプロセス制御システム。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】本願発明は、プロセス制御システムに関し、より詳しくは、プロセス制御システム内で地理的に分散された物理的な位置からアクセスおよび更新が可能なコンフィギュレーション・データベースを有するプロセス制御コンフィギュレーション・システムとしての、コンフィギュレーション・データベース・システム、分散型のコンフィギュレーション・データベース・システム、およびプロセス制御システム、並びに、本システムに用いるデータベース・サーバ、更には、コンフィギュレーション・データを格納および使用する方法、データベース・コンポーネントへのアクセスを提供する方法、およびデータベースに格納されたコンポーネントへの並列アクセスを提供するためにデータベース・サーバを使用する方法に関する。

## 【0002】

【従来の技術および発明が解決しようとする課題】本願は、1999年10月18日に出願された仮出願第60/160,104号の「プロセス制御システム内で分散された物理的な位置からのコンフィギュレーション・データベースに対するアクセスおよび更新」に基づく出願である。

【0003】プロセス制御システムは、化学、石油、又はその他のプロセスに利用されるもののように、概して、アナログおよび／若しくはデジタル・バス又はその他の通信線又はチャネルを介して、少なくとも1つのホスト・デバイス又はオペレータ・ワークステーションと一又は複数のフィールド・デバイスとに通信可能に結合される少なくとも1つのプロセス・コントローラを備えている。前記フィールド・デバイス、即ち、例えば、バルブ、バルブポジショナー、スイッチ、および送信機（例えば、温度、圧力、および流量センサ）等は、バルブの開閉、およびプロセス・パラメータの測定の如きプロセスにおける機能を実行する。プロセスのランタイム中にあっては、前記プロセス・コントローラは、フィールド・デバイスおよび／又は入力／出力（I/O）装置を介した該フィールド・デバイスに関するその他の情報によってなされたプロセス計測を示す信号を受信し、制御ルーチンを実行するためにこの情報を使用して、当該プロセスの動作を制御すべく、前記入力／出力装置を介してバス又はその他の通信チャネルを通じて前記フィールド・デバイスへ送信される制御信号を生成する。前記フィールド・デバイスおよび前記コントローラからの情報は、プロセスの現在の状態を見ること等のプロセスに関するどのような要求機能でもオペレータが行えるように、前記オペレータ・ワークステーションにより実行される一又は複数のアプリケーションについて、概してランタイム中に利用できるようになされている。加えて、ホスト・デバイス、ワークステーション、およびラップトップ・コンピュータ等の如きユーザ・インタフェースで実

行されるコンフィギュレーション・アプリケーションは、プロセスの操作を修正するため、プロセスを設定するため、プロセスのコンフィギュレーションを見るため、およびプロセス・コンフィギュレーションを記録するため等に利用される可能性がある。

【0004】概括して言えば、プロセス制御システムは、該プロセス制御システム内でハードウェアおよびソフトウェア・コンポーネントの各々に関連するコンフィギュレーション情報を格納するコンフィギュレーション・データベースを用いるように構成されており、プロセスにおける異なるデバイスおよびコントローラの如きハードウェア・コンポーネントが、物理的に相互接続され、また、制御モジュールおよび通信モジュール等の如き異なるソフトウェア・コンポーネントが、プロセス制御を実行するために、プロセス制御システム内における異なるデバイスによって連携および実行される。場合によっては、前記コンフィギュレーション・データベースは、オブジェクトとしてプロセス制御システムの各々異なる論理的なコンポーネントのためにコンフィギュレーション・オブジェクト又はコンフィギュレーション・コンポーネントを格納するオブジェクト指向データベースである。前記コンフィギュレーション・データベースは、例えば、幾つか又は全てのハードウェアおよびソフトウェア・コンポーネントのためのオブジェクト・テンプレートを格納するライブラリを備えることが可能である。そして、これらのテンプレートは、プロセス制御システム内で実際に用いられている例えばハードウェア又はソフトウェア・コンポーネントのためのコンフィギュレーション・オブジェクトを生成するために使用される。前記コンフィギュレーション・データベースは、また、プロセス制御システムの物理的なコンポーネントがセットアップされ、分配され、そして相互接続されるような、セットアップ又は物理ネットワーク・セクションを備えることが可能である。場合によっては、前記コンフィギュレーション・データベースは、また、コントローラ、フィールド・デバイス、および該コントローラおよび／又はフィールド・デバイスにおいて実行される制御モジュール又は制御ルーチンを用いてどの様に制御がなされるかを定義する制御セクションを備える。前記プロセス制御システムのコンフィギュレーションにおいて、例えばユーザ・インタフェース又はワークステーションで実行されたコンフィギュレーション・ルーチン又はアプリケーションは、前記プロセス制御システムの実際のコンフィギュレーションを反映してコンフィギュレーション・データベースを生成又は修正するために使用される。このコンフィギュレーション・アプリケーションは、概して、前記プロセス制御システムに関連するデバイスおよびコントローラ等を設定すべく前記コンフィギュレーション・データベース内の情報を使用し、また、デバイス又はソフトウェア・コンポーネントが前記

システムに加えられる時等のような如何なるコンフィギュレーション動作が行われた後であっても、新しいコンフィギュレーション情報を前記コンフィギュレーション・データベースに格納するようになっている。

【0005】また、前記コンフィギュレーション・データベースは、前記プロセス制御システムに接続されたユーザ・インタフェースを介して、ユーザに前記プロセス制御システムの現在のコンフィギュレーションを表示するために一般に用いられる。過去においては、プロセス制御システムのユーザ・インタフェース、コンフィギュレーション・データベース、およびコントローラの幾つかは、ローカル・エリア・ネットワーク（LAN）を形成すべく、イーサネット（登録商標）・バスの如き専用バスによって通信可能に相互接続されていた。専用のイーサネット・バスが大きい帯域幅を有し、また、該イーサネット・バス上を送信される任意の特定の信号又はデータ要求をLAN内であまり長い距離に亘って往来させる必要がないので、前記ユーザ・インタフェースのうちの1つによる前記コンフィギュレーション・データベースとの通信は、これらのシステムにおいて相当容易且つ高速である。その結果、前記ユーザ・インタフェースにおいて実行されるコンフィギュレーション表示ルーチンは、ユーザが前記プロセス制御システムのコンフィギュレーションに関連する情報を示すか又は要求する都度、概して、前記コンフィギュレーション・データベースにアクセスして、コンフィギュレーション情報を取り出す。この取り出された情報は、それから、イーサネット・バスを通じて送出されて、前記ユーザ・インタフェース上にユーザに表示される。前記専用のイーサネット・バスの速度（又は大きい帯域幅）のため、複数のユーザが、前記コンフィギュレーション・データベースに比較的同時にアクセスすることができ、また、前記プロセス制御システムのコンフィギュレーションに関連する同一の又は異なるコンフィギュレーション・データを見ることができる。同様に、異なるユーザが、プロセス制御システムの異なる部分を再設定することができる。なぜなら、生成された如何なる新しいコンフィギュレーション・データであっても、前記専用のイーサネット・バスを介して比較的短い時間で、前記コンフィギュレーション・データベースに直接的に提供され得るからである。さらに、前記LANに接続されたこれらのデバイスだけが前記コンフィギュレーション・データベースにアクセスすることができるので、また、前記LANが単一のプロセスの位置に概して限られるので、多数のユーザ・インタフェースがコンフィギュレーション・データベースに同時にアクセスすることを可能にする必要は余りない。

【0006】近年、しかしながら、幾つかのプロセス制御システムのサイズに対する一般的な増加があり、また、より大きい又は離隔された地理的なエリアに亘って拡散又は分配されたプロセス制御ネットワーク用に、コ

ンフィギュレーション情報を統合したいという要望があった。例えば、或るユーザは、異なるプロセス・サイトに関連するコンフィギュレーション・データを結合することを望み、当該プロセス・サイトは、異なる郡、州、又は更には異なる国に存在する可能性がある。それによって、第1のサイトにおけるオペレータに対して、第2のサイトについてのコンフィギュレーション情報へのアクセスおよび該コンフィギュレーション情報の閲覧を許容し、更には、第1のサイトから第2のサイトに影響を与えるコンフィギュレーション動作を行うことさえ許容することも可能である。別の例では、ユーザが、海洋を遠く離れ、精油プロセス制御システムを備えたプラットフォーム上での油井を、本土に存在する多数のコントローラおよびユーザ・インタフェース等を用いて統合したいと要望するかも知れない。この場合、精油サイトにおけるユーザに対して、前記プラットフォームまで実際に飛んで行くことなしに、油井プラットフォームでのデバイスの再設定又は設定（コンフィギュレーション）に影響を与えることを可能とすることが望ましい。つまり、現在一般的に必要であるように、油井プラットフォームのターミナルにユーザ・インタフェースを接続し、油井コントローラを再設定するのである。さらに、複数のプロセス制御サイトの統合にあつては、これらのサイトのコンフィギュレーション・データベースに同時にアクセスすべく用いられ得るユーザ・インタフェースの数はかなり増加する。

【0007】複数の地理的に離隔したサイトを統合する際には、しかしながら、それは、距離に起因して、イーサネット・バスの如き共通の専用バスを使用する異なるサイトを接続することが、実際には不可能である。しかしながら、衛星、セルラー、若しくはその他の種類の無線接続、又はインターネット若しくはT1回線の如き広域ネットワーク接続を使用すること、プロセス制御システムの異なるサイト間の通信接続を確立することは可能であり、それによって、プロセス制御ネットワークの地理的に離隔した部分の統合を可能にする。しかし、衛星、セルラー、およびその他の広域無線通信接続は、概して、その使用に際して非常に高価であり、これらが、イーサネット・バスの如き専用バスと比較して限られた帯域幅のみを提供することができるように一般に共有される。同様に、インターネット、T1回線、およびその他の共有広域ネットワークは、限られた帯域幅又はスループットだけを提供するので、専用のイーサネット・ローカル・エリア・ネットワーク・バスと比較して、概してデータ転送が非常に遅い。また、異なるサイト間の地理的な距離と、例えばデータ・パケットの認証を用いるサイト間の安全な保証された通信を提供するために必要なこととは、IF、TCP、およびUDP通信プロトコルにおいて提供されるように、サイト間の通信の遅れを更に増大する。

【0008】プロセス制御システムの地理的に多様なサイト間の通信における遅れと、コンフィギュレーション・データベースにアクセスするユーザ・インタフェース又はその他のデバイスの数の増加との結果として、異なるサイトの全てにおけるユーザによるアクセスのような、また、現状のコンフィギュレーション・データベースのアクセス手順を用いて異なるサイトにおけるユーザによる変更のような、地理的に多様なサイトの各々でのデバイスを含むプロセス制御システム全体のためのデータを格納するコンフィギュレーション・データベースを提供することは難しい。特に、コンフィギュレーション・データベースがメイン・サイトにあるならば、リモート・サイトにおけるユーザが、遅いリンク又はリモート・リンクを通じてリフレッシュ単位でコンフィギュレーション・データの全てをダウンロードしなければならない、甚大な時間をとられることになる。また、コンフィギュレーション・データベースにアクセスしているユーザの数の顕著な増加は、コンフィギュレーション・データベース内の如何なる情報へのアクセス時間をも容認できないレベルにまで増大することになる。同様に、プロセス制御システムを再設定する際、メイン・サイトおよびリモート・サイト間の通信の遅れは、2人の異なるユーザに対して、混乱およびエラーに導く、プロセス制御システムの同一のコンポーネントの再設定を行わせる。このように、例えば、リモート・サイトにおけるユーザは、メイン・サイトでのコンフィギュレーション・データベースからのプロセスの現在のコンフィギュレーションを要求し、得ることが可能であり、リモート・サイトでのプロセス制御システムの部分を再設定し、それから、メイン・サイトでのコンフィギュレーション・データベースに新しいコンフィギュレーション・データを送信することが可能である。一方、しかしながら、メイン・サイトにおけるユーザは、プロセス制御システムの同一部分を再設定することが可能である。なぜなら、メイン・サイトでの通信に比較的遅れがないからであって、リモート・サイトにおけるユーザは、エラーを導く、格納のためにコンフィギュレーション・データベースへリモート・サイトで生成された新しいコンフィギュレーション・データを送ろうとする前に、この新しいコンフィギュレーションをコンフィギュレーション・データベースに格納することが可能である。

【0009】場合によっては、異なるサイト間での遅く小さい帯域幅の通信リンクを提供することでさえ、不可能であるか又は実用的ではないかも知れない。この場合、リモート・サイトのコンフィギュレーションは、リモート・サイトでなされなくてはならず、従ってメイン・サイトでコンフィギュレーション・データベースにアップロードされなければならない。しかし、このオフラインのコンフィギュレーションは、異なるユーザが同時又は略同時にリモート・サイトを再設定しようとする場

合に、又はリモート・サイトを既に再設定した別のユーザが該リモート・サイトに既になされた変更を反映するためにコンフィギュレーション・データベースを更新する前に、最初のユーザが再設定しようとした場合に、問題を引き起こすことがあり得る。

【0010】本願発明は、プロセス制御システムの異なる位置が、小さい帯域幅若しくは遅い（即ち、遅延された）通信回線を介して通信可能に接続されるか、又は例えばモデム接続を介して間欠的に通信可能に接続されるときであっても、異なる位置の各々におけるユーザに、タイムリーで、安全で、且つ非矛盾したコンフィギュレーション・データベースの変更を可能とするように、共通のコンフィギュレーション・データベースを用い、プロセス制御システムの多数の地理的に離隔した位置を統合することを可能にするための、コンフィギュレーション・データを格納および使用する方法、データベース・コンポーネントへのアクセスを提供する方法、およびデータベースに格納されたコンポーネントへの並列アクセスを提供するためにデータベース・サーバを使用する方法、並びにコンフィギュレーション・データベース・システム、分散型のコンフィギュレーション・データベース・システム、およびプロセス制御システム、並びに、本システムに用いるデータベース・サーバを提供することを目的とする。

【0011】

【課題を解決するための手段】本願発明の1つの観点によれば、共有されたコンフィギュレーション・データ、例えば、複数の位置に関連する又は複数の位置に適用可能なコンフィギュレーション・データが、遅い又は小さい帯域幅の又は複数の通信回線を介してその他の位置の各々によって評価可能なデータベースに格納されるように、また、特定の位置だけに関連するデータが、その位置でコンフィギュレーション・データベースに格納されるように、プロセス制御システムの異なる地理的な部分上又は部分を介して分散されたコンポーネントを備える分散型のコンフィギュレーション・データベースを提供する。このように、特定の位置で最もアクセスされると思われるコンフィギュレーション情報（例えば、その位置に関連するコンフィギュレーション情報）は、その他の位置に関連するコンフィギュレーション情報が遅い又は小さい帯域幅の通信回線を通じてアクセスされ得る一方、ローカル・エリア・ネットワークを通じてアクセス可能である。その結果、共有通信データと異なる位置に関連するデータだけは、遅い又は小さい帯域幅の通信回線を通じて送信される必要がある。

【0012】プロセス制御システム内でユーザによって見られているコンフィギュレーション・データが古くなることを防ぐ目的から、各位置でのコンフィギュレーション・データベース・アクセス・システムは、そのデータを現在見ているユーザの各々に、コンフィギュレーション

ョン・データベースになされた変更を自動的に送信することが可能であって、あたかもコンフィギュレーション・データベース内にそのデータが存在するかのように、ユーザがコンフィギュレーションの状態を見ることができる。1つの実施形態において、コンフィギュレーション・データベースの各部分は、コンフィギュレーション・データベースへの読み出し回数を低減するように、コンフィギュレーション・コンポーネントの伝達をそのデータを現在見ているユーザの各々に又は予約者に調整する共有キャッシュを使用するデータ・アクセス・ルーチンを有しており、コンフィギュレーション・データベースからデータにアクセスする速度を増大する。この共有キャッシュは、コンフィギュレーション・データベース内でコンポーネントのロックと予約とを可能にするロック・メカニズムを組み込むことが可能である。望むならば、たとえ、その情報のマスター・コピーを格納するコンフィギュレーション・データベースへの通信回線が、切断されるか、又は利用できなくなるとしても、各々のユーザによって得られたコンフィギュレーション情報は、ユーザがコンフィギュレーション情報を見ることができるよう、ローカルに格納されることが可能である。

#### 【0013】

【発明の実施の形態】以下、本願発明の実施形態を図面を参照しながら説明する。図1において、プロセス制御システム10は、双方向チャンネルからなる第1の上り回線／下り回線チャンネル20と第2の上り回線／下り回線チャンネル22とを有する衛星18によって構成された衛星通信回線16を介して通信可能に接続された地理的に離隔した第1および第2のサイト（又は位置）12、14を備えている。第1のサイト12（ここでは、ローカル又はメイン・サイト12と称す）は、例えば、イーサネット又はその他の任意の好ましいバス又は通信線である専用のローカル・エリア・ネットワーク・バス（LANバス）34を介してユーザ・インタフェース28に接続された、2つのプロセス・コントローラ24、26と、コンフィギュレーション・データベース30と、データ・サーバ32とを備えている。ユーザ・インタフェース28は、任意の好ましい種類のホスト・ワークステーション又はコンピュータであり、例えば、任意の種類のパーソナル・コンピュータ、ラップトップ・コンピュータ等である。この際、コンフィギュレーション・データベース30は、スタンド・アローン型のデータベース装置、又は例えばユーザ・インタフェース28若しくはデータ・サーバ32の如きその他の任意のデバイスに統合されていてもよい。データ・サーバ32は、データ・サーバ32が第2のプロセス制御サイト14へ遅い又は小さい帯域幅の通信回線16を通じて通信するために使用するアンテナ39を備える。なお、ここでは、第2のプロセス制御サイト14をリモート・サイト14と称す。

【0014】前記プロセス・コントローラ24、26の各々

は、一又は複数の入力／出力（I/O）装置36に接続されている一方、該I/O装置36は、従来の420ミリアンペアのデバイス38又は任意の好ましい種類のフィールド・デバイス（F.D）（例えば、「HART（商標）」、「PROFIBUS（商標）」、「アクチュエータ・センサ・インタフェース（AS-Interface）」、「WORLDFIP（商標）」、「Device-Net（商標）」、「CAN」、「FOUNDATION（登録商標）Fieldbus（以下に「Fieldbus」と称す）等のような高性能フィールド・デバイス）40に接続され得る。勿論、前記デバイス38、40は、プロセス制御システム10内で任意の制御、測定、又はその他の機能を行い得る任意の好ましい種類のデバイス（例えば、センサ、送信機、バルブ、ファン、ミキサー、ポジショナー等）とすることが可能である。前記プロセス・コントローラ24、26は、任意の公知の又は好ましい種類のI/O装置36と、前述した何れかに関連する通信プロトコルとを用いるフィールド・デバイス38、40と通信することが可能である。概括して言えば、プロセス・コントローラ24、26は、その一例として、本願出願人であるフィッシャー・ローズマウント・システムズ・インコーポレイテッド社から販売されている「DeltaV（登録商標）」コントローラとすることが可能であり、各々は、プロセッサと、データを格納するメモリと、プログラムと、メイン・サイト12でプロセス（プロセス制御システム）10の制御を実行すべく用いられる制御ルーチン（例えば、プロセス制御モジュール）とを備えている。概括して言えば、前記プロセス・コントローラ24、26は、フィールド・デバイス38、40から信号を受信し、プロセス制御ルーチンを実行して、それによってプロセス（プロセス制御システム）10の制御を実行するために前記デバイス38、40に制御信号を与える。

【0015】同様に、前記リモート・サイト14は、専用のローカル・エリア・ネットワーク・バス（LANバス）58（例えば、イーサネット又はその他の好ましいバス又は通信線）を介してユーザ・インタフェース52、53に接続された、プロセス・コントローラ（コントローラ）50と、コンフィギュレーション・データベース54と、データ・サーバ56とを備えている。前記コントローラ50は、2つの高性能フィールド・デバイス（F.D）62にI/O装置60を介して接続されているように図示してあるが、その他の任意の個数又は種類のフィールド・デバイスに接続され得るものである。前記データ・サーバ56は、遅い又は小さい帯域幅の回線16を介してデータ・サーバ32と通信するためにアンテナ64を利用し、これによって、ローカル・サイト12とリモート・サイト14との間における通信を達成する。リモート・サイト14内の各デバイスは、ローカル・サイト12内の対応する各デバイスと同一又は同様のものであり得、リモート・サイト14においてプロセス制御および報告動作を行う。同様に、任意の好ましい方法でプロセス制御およびコンフィギュレーション動

作を行うために、サイト12, 14の何れか一方又は両方において、その他のデバイス又は更にその他の個数のデバイスが接続できるということである。実際、前記サイトのうちの1つ（例えば、リモート・サイト14）は、望むならば、ユーザ・インタフェースのような単一のデバイスであり得る。前記データ・サーバ32は、直接コンフィギュレーション・データベース30に接続され得、また、同様に、前記データ・サーバ56は、イーサネット・バスを介して接続される代わりに、直接コンフィギュレーション・データベース54に接続され得る。

【0016】このように、前記データ・サーバ32, 56は、2つのLANバス34, 58間の通信接続を提供するために動作するものであって、これによって、ローカル・サイト12およびリモート・サイト14内のデバイス間の通信を可能にする。前記通信回線16が衛星回線として図示されているが、代わりに、例えば、セルラー回線、モデム、電話回線、インターネット回線、その他の任意の無線回線、又は広域若しくは共有ローカル・エリア・ネットワーク回線等のその他の任意の通信回線が用いられ得る。勿論、任意の好ましい通信方法が、前記回線16を介したデータ通信を提供すべく、通信回線16内で使用されることが可能である。従って、任意の通信プロトコル（例えば、IP, TCP, 又はUDPプロトコル）が使用されることが可能であり、例えば、任意の変調技術、エラー符号化技術等が、例えばスペクトル拡散技術を含む回線16を介して通信を行うことに使われ得る。好ましくは、或る種のデータ認証方法が、雑音又はその他の干渉が存在する中で安全で信頼できる通信を保証するために通信回線16で使用する。望むならば、本願発明の譲受人に譲渡され、また、ここで参照することによってここに明白に取り込まれる、1999年10月15日に出願された「遅延認証通信および警報管理」という米国特許出願No. 09/418747号に記述されたデータ認証方法又は技術を、通信回線16を介した通信を達成することに使用することも可能である。概括して言えば、前記通信回線16の使用は、互いに統合されるべき地理的に離隔したプロセス制御サイト又はシステムを、1つのプロセス制御ネットワークとして構成することを可能にする。ここで、1つのサイト内のデバイスは、制御動作およびコンフィギュレーション動作を行うべく、他のサイトにおけるデバイスと通信することができる。

【0017】残念なことに、前記通信回線16を通じた通信は、概して、前記サイト12, 14間の距離のために、LANバス34, 58を通じた通信よりも非常に遅いものである。さらに、前記通信回線16は、専用のLANバス34, 58と比較して小さい帯域幅を有するかもしれない、該LANバス34, 58を通じた通信よりも一般に雑音およびその他のエラーに影響され易い。これらの要因は、一般に、プロセス制御システム10内に通信のボトルネックの原因を、通信回線16に作ることに繋がる。

【0018】本願発明によれば、プロセス制御システム10が設定される方法に関連するコンフィギュレーション・データは、コンフィギュレーション・データベース30, 54の一方又は両方に格納される。また、一又は複数のユーザ・インタフェース28, 52, 53、又はその他のデバイスは、コンフィギュレーション・アプリケーションを格納し、また、オペレータの要求によって、該アプリケーションを実行することが可能である。このコンフィギュレーション・アプリケーションは、コンフィギュレーション情報（例えば、プロセス制御ネットワーク10内のデバイスのコンフィギュレーション、ソフトウェア・モジュール等に関連する情報）を得るコンフィギュレーション・データベース30, 54のうちの1つ以上にアクセスすることが可能であり、ユーザは、得られたコンフィギュレーション情報を画面上に表示させることが可能である。前記コンフィギュレーション・アプリケーションは、また、ユーザに対して、新しいデバイス、ソフトウェア・コンポーネント、又はその他のコンポーネントを前記システムへ付加し、前記システム内のデバイス間の新しい通信を提供し、前記システム内の既存のコンポーネントを変更する等して、前記プロセス制御システム10の再設定を可能とする。別の例では、第1のデータベース・サーバ（例えば、データ・サーバ32）において実行されたコンフィギュレーション・アプリケーションは、更に異なるリモート位置（図1においては図示せず）にこの情報を提供するために、前記遅い回線16を介してもう1つのデータベース・サーバ（例えば、データ・サーバ56）から情報にアクセスすることが可能である。勿論、本願出願人であるフィッシャー・ローズマウント・システムズ・インコーポレイテッド社から販売されている製品「DeltaV」を備えた前記コンフィギュレーション・アプリケーションのような、任意の好ましいプロセス制御コンフィギュレーション・アプリケーション又はルーチンを使用することが可能である。特に、前記「DeltaV」のコンフィギュレーション・アプリケーションは、プロセス制御のコンフィギュレーション情報をウィンドウズ（登録商標）・エクスプローラー式のツリー構造で表示し、ユーザ又はオペレータがコンポーネントをコンフィギュレーション・ツリー内で一方から他方へドラッグ・アンド・ドロップすることによって、前記システム10を再設定することが可能であって、これにより、異なるデバイス内のこれらのコンポーネントをロードするか、プロセスの異なるエリアでプロセス制御コンポーネントの配置を示すか、異なるコントローラ、バス、若しくはI/Oデバイス等へデバイスの接続を示すか、又は異なるデバイス内でソフトウェア・コンポーネントをロードする。ドーブ（Dove）らの米国特許出願第5,838,563号（プロセス制御環境を最設定するシステム）、ニクソン（Nixon）らの米国特許出願第5,828,851号（標準デバイスおよび非標準デバイスの標準プロトコル制御を用い

たプロセス制御システム)、1996年4月12日に出願されたニクソンらの米国特許出願第08/631519号(ネットワークへのデバイスの接続を自動的に検出する方法および装置を有するプロセス制御システム)、および1996年4月12日に出願されたドーブの米国特許出願第08/631458号(プロセス制御環境の設定を支援するシステム)は、全て本願発明の譲受人に譲渡されており、また、ここで参照することによってここに明白に取り込まれる。これらは、ユーザに対して、プロセス制御ルーチンおよびコンポーネントを視覚的に作成すること、プロセス制御システム内のデバイスの制御を自動検出すること、およびプロセス制御システム内のデバイスの制御を提供することを可能にするプロセス制御コンフィギュレーション・アプリケーションについて記述したものである。勿論、その他の任意のグラフィック・コンフィギュレーション・アプリケーションおよび非グラフィック・コンフィギュレーション・アプリケーションを含むその他の任意のコンフィギュレーション・アプリケーションが、前記コンフィギュレーション・データベース30, 54にアクセスするために、ユーザ・インタフェースにおいて又はサーバ・データベース又はその他のデバイスにおいて実行され得る。

【0019】図2は、プロセス制御システムのコンフィギュレーションを表示するために用いられることが可能であり、また、プロセス制御システムのコンフィギュレーションへの変更を視覚的に特定するための用いられることが可能である、プロセス制御のコンフィギュレーション・ツリー65の一例を示している。概括して言えば、図2のコンフィギュレーション・ツリー65は、「トレーシー・アイランド」と名付けられたプラントのコンフィギュレーションに関連する情報を図示および提供している。前記トレーシー・アイランドのコンフィギュレーション・システムは、例えば、異なるデバイスの設定、デバイスの定義、ソフトウェア・コンポーネント等のテンプレート(例えば、制御モジュール)、又は前記トレーシー・アイランドのプラントで使用されるその他のオブジェクトを有することが可能であるライブラリ・セクション66を有している。これらのテンプレートは、前記ツリー65において示されていないが、ライブラリ・アイコン66を選択する(例えば、ダブル・クリックする)ことによって、これらを表示することが可能である。前記コンフィギュレーション・ツリー65は、また、前記デバイス(例えば、コントローラ、I/O装置、フィールド・デバイス、ユーザ・インタフェース等)が前記プロセス内で物理的に配置されるようなシステム・コンフィギュレーション・セクション67を有することが可能である。前記トレーシー・アイランドのプラントのための図示のシステム・コンフィギュレーションは、レシピ(例えば、異なる製品を製造するプラントにおいて行われるプロセス)、セットアップ情報、および制御方法を有してい

る。図示の如く、前記制御方法のアイコン又はコンポーネントは、未割当のI/Oリファレンス、設備、および「Area\_A」および「Areal」と名付けられた2つのエリア(物理的なエリア)を含む多数の子コンポーネントを有している。前記「Area\_A」は、「Module1」制御モジュールがトレーシー・アイランドのプラントの「Area\_A」の部分内でロードされたことを図示する「Module1」子コンポーネントを有している。前記システム・コンフィギュレーションは、また、非稼働中のコントローラと制御ネットワークのアイコンとのリストを有する物理ネットワークを備えている。前記制御ネットワークのアイコン又はコンポーネントは、一又は複数の稼働中のコントローラ(コントローラ「CTRL1」のみを図示)と、該コントローラ「CTRL1」の操作に関連付けられたユーザ・インタフェース(「Governmint」と名付けられている)とを有している。各コントローラは、これに取り付けられた一又は複数のI/O装置を備えることが可能であり、また、各I/O装置は、これに取り付けられ、前記ツリー65内で前記コントローラに通信可能に結合されるべき一又は複数のフィールド・デバイスを備えることが可能である。しかし、これらのコンポーネントは、図2には示されていないが、例えば、「CTRL1」のアイコンがユーザによって選択された場合に表示されるようになっている。

【0020】なお、図2に図示されたコンフィギュレーション・コンポーネントは、親子関係を有し、ここで、例えば、前記システム・コンフィギュレーション・セクション67は、レシピ、セットアップ、制御方法、および物理ネットワークの子供を示すことを注記しておく。同様に、前記制御方法は、非割当のI/Oリファレンス、設備、「Area\_A」、および「Areal」の親である。また、「Area\_A」は、「Module1」コンポーネントの親である。勿論、子コンポーネントの各々は、更なる子供を有することが可能であり、そのため、前記ユーザは、オブジェクトがどのデバイスに格納されているか、どのデバイスがどの物理的なゾーンに配置されているか、また、異なるデバイスがどの様に通信可能に相互接続されているかについて見るために、コンフィギュレーション・ツリー65の如何なるレベルでも見る事が可能である。通常、図2の階層は、親コンフィギュレーション・コンポーネントの次にプラス(+)のシンボルを配置することによって、非表示の子コンポーネントの存在を示している。また、その子供を含む選択されたコンポーネントに関する情報は、図2の制御方法コンポーネントで図示されるように、画面の右側に表示される。勿論、これらのコンポーネントに関する更なる情報は、図2の階層内でこれらのコンポーネントを選択することによって得られる。

【0021】図2にコンフィギュレーション階層の一例を示すように、プロセス制御ネットワークのための前記



コンフィギュレーション階層は、その他のデバイスおよびモジュールを示すことができ、また、その他の従属関係をとることができる。また、プロセスは、多数の物理サイト、および／又はゾーンに分割されることが可能である。そして、これらのサイト又はゾーンの間は、前記階層において示されるような、自体に関連するコントローラ、ユーザ・インタフェース等を有することが可能である。勿論、ユーザは、初めに図2のツリー65の一部を見るかもしれないが、それからコンフィギュレーション・ツリー65（例えば、「AreaA」）内のコンポーネントを選択して、コンフィギュレーション・ツリー（階層ツリー）65内へ更に掘り下げていくことが可能である。この選択によって、階層ツリー65を提供しているコンフィギュレーション・アプリケーションに対して、選択されたオブジェクトの子供にアクセスして、表示画面上にこれらの子供を表示させる。場合によっては、前記コンフィギュレーション情報は、オブジェクト指向データベース内にオブジェクトとして格納されることが可能であり、また、前記データベース内のオブジェクトは、前記コンフィギュレーション・アプリケーションによって表示された前記コンフィギュレーション・ツリー内に示されるものと同様の関係を有することが可能である。

【0022】図2の階層ツリー65において図示されるコンフィギュレーション・オブジェクト又はコンポーネントは、図1のコンフィギュレーション・データベース30, 54に格納されることが可能であり、また、コンフィギュレーション・データベース30, 54からアクセスされることが可能であると言える。前記コンフィギュレーション・データベース30, 54（例えば、任意の好ましい種類のオブジェクト指向データベース）に格納される情報は、任意の好ましい方法で格納されるかもしれないが、また、図2に示した関係に関連して又は該関係と同様の関係を有している。

【0023】従来技術のプロセス制御ネットワークには、概して、離隔されたサイト（例えば、遅い又はリモート通信回線を介して接続されたサイト）がない。このように、各プロセス制御システムは、概して、コンフィギュレーション・データをそのシステム内のデバイスのコンフィギュレーションだけに関連するそれ自身のコンフィギュレーション・データベースを備えている。さらに、このコンフィギュレーション・データベースは、概して、ユーザ・インタフェース又は前記プロセス制御システムのLANに接続されたその他のデバイスの全てによってアクセスされることができ、LANバスの大きい帯域幅と、LAN内のユーザ・インタフェースおよびコンフィギュレーション・データベースの間の比較的短い距離のために、非常に速くアクセスされ得る。このように、例えば、従来技術のシステムにおいて、前記サイト12, 14の各々は、前記サイト（即ち、サイト12又はサイト14、

但し、両方ではない）のうちの1つ内のデバイスだけに関連する情報を格納するそれ自身のコンフィギュレーション・データベースを備えている。これらの場合、前記サイト12, 14間の通信回線16を提供することは、これらのサイト間の通信を可能にすることであって、1つのサイトのためのコンフィギュレーション・データベースは、その他のサイト内のデバイスのコンフィギュレーションを反映せず、図2のそれのような、コンフィギュレーション・アプリケーションが、別々のコンフィギュレーション・データベースに格納されたコンフィギュレーション・コンポーネントを信号のコンフィギュレーション階層又はツリーに統合することはない。

【0024】また、従来技術のコンフィギュレーション・アプリケーションは、ユーザが前記コンフィギュレーション階層内で更なる情報のために掘り下げる操作又はその他の要求を行う都度、ユーザによって要求されている情報を得るためにコンフィギュレーション・データベースに概してアクセスするものである。残念なことに、実際のデータをコンフィギュレーション・データベースから読むことは、データを引き出すプロセスの最も遅い部分である。従って、多数のユーザの各々がコンフィギュレーション階層を見て、前記コンフィギュレーション・データベースに読み書きすることは、単一のデータベースからこの情報にアクセスすることができることを過度に遅らせることになる。このことは、より多くのユーザが一又は複数のリモート通信回線を介して前記システムに加えられるときに悪化し、前記コンフィギュレーション・データベースにアクセスすることができるユーザの数を増大する。同様に、データをこれらの離れて配置されたユーザのために引き出すことは、リモート回線（例えば、図1の通信回線16）に関連する通信におけるボトルネックによって、更に遅らせられる。

【0025】前記データベース30, 54および前記プロセス制御システム10内の通信は、前記プロセス制御システム10についてのコンフィギュレーション情報を前記ローカル・サイト12およびリモート・サイト14の一方又は両方における多数のコンフィギュレーション・アプリケーションに提供するための多くの方法によって、また、前記リモート回線16を通じた通信量を低減する方法によって、構成され得、前記プロセス制御システム10の全て又は任意の部分の統合図を提供し、また、前記コンフィギュレーション情報をプロセス制御システム10内の任意の位置から変更することを許容する。最初のケースでは、コンフィギュレーション・データベース（例えば、図1のコンフィギュレーション・データベース30）のうちの1つは、全てのコンフィギュレーション項目のマスター・コピー又はその中のコンポーネントを格納するマスター・コンフィギュレーション・データベースに指定される。その他のコンフィギュレーション・データベース54は、ブリーフケース・データベースと称され、該ブリー

フケース・データベースは、必要に応じて、マスター・コンフィギュレーション・データベース30内の情報の幾つか又は全てにアクセスし、このデータを前記通信回線16を介してダウンロードし、また、リモート・サイト14におけるユーザ・インタフェース又はその他のデバイスにそのデータを見て、変更することを可能にする。変更されたデータは、その他のユーザが前記マスター・コンフィギュレーション・データベース30を介して当該データにアクセスをすることを保証する通信回線16を介して前記マスター・コンフィギュレーション・データベース30にアップロードされ得る。この場合、必要とされたコンフィギュレーション・データの全てが前記ブリーフケース・データベース54にダウンロードされることができ、また、リモート・サイト14においてこのデータベース54からアクセスされ得るので、前記遅い又は小さい帯域幅の通信回線16を介した前記マスター・コンフィギュレーション・データベース30への個々のユーザ・インタフェースによる呼び出しは低減される。さらに、ブリーフケース・データベース52に多数のコンフィギュレーション・コンポーネントを一度にダウンロードするプロセスは、通信回線16の使用に関しては、少しずつこの情報をダウンロードしようとするよりも効率的である。

【0026】別のケースでは、コンフィギュレーション・データベース54は、マスター・コンフィギュレーション・データベース30の状態を反映するミラー・データベースであることが可能である。ミラー・データベース54は、それが生成されるとき、マスター・コンフィギュレーション・データベース30の状態を反映するためにマスター・コンフィギュレーション・データベース30からデータの全てをコピーする。この方法によれば、例えばミラー・データベース54内のルーチンは、遅い通信回線16を介してマスター・コンフィギュレーション・データベース30に定期的にアクセスし、ミラー・コンフィギュレーション・データベース54においてなされた変更をマスター・コンフィギュレーション・データベース30に提供することによって、また、マスター・コンフィギュレーション・データベース30になされた変更をミラー・コンフィギュレーション・データベース54にコピーすることによって、マスター・コンフィギュレーション・データベース30と整合させる。ここでは、リモート・サイト14におけるユーザが、遅い回線16を通じた個々のユーザによってなされたデータの呼び出しの回数を減らすか又は該呼び出しを無くすミラー・コンフィギュレーション・データベース54にアクセスする一方、メイン・サイト12におけるユーザは、マスター・コンフィギュレーション・データベース30にアクセスする。

【0027】更に別のケースでは、コンフィギュレーション・データの異なる部分は、分散型のコンフィギュレーション・データベースを構築するために、コンフィギュレーション・データベース30、54に格納される。ユー

ザ・インタフェース又はその他のデバイスにおいて実行されるコンフィギュレーション・アプリケーションは、ユーザが、見ること、又はこのデータが元々格納されて、これらのコンポーネントへの変更を予約するデータベースから、操作することを望むコンフィギュレーション・コンポーネントにアクセスする。1つの実施形態において、予約(subscription)回線又はスレッドは、各々のユーザ又はクライアントと、ユーザ又はクライアントがコンフィギュレーション・データを引き出すためにアクセスしているデータベースの各々との間で確立される。コンフィギュレーション・データベース54に格納され、ユーザ・インタフェース53(これらの両方ともリモート・サイト14にある)によって予約されたコンフィギュレーション・データに対しては、前記スレッドは、通信回線16を介した通信を必要としない。しかし、コンフィギュレーション・データベース30に格納され、ユーザ・インタフェース53(これらは異なるサイトにある)によって予約されたデータに対しては、前記スレッドは、通信回線16を介した通信を必要とする。その結果、前記サイトにある又はこのデータが最もアクセスされそうであるその他の位置にあるコンフィギュレーション・データベース内のコンフィギュレーション・データの各部分を格納することが望ましい。この方法の詳細は後で更に説明する。

【0028】ここに記述された技術は、多くの雑音が存在する中(多数のリトライを引き起し得る)で、遅い又はリモート通信回線を通じてリモート・システムにコンフィギュレーションの変更をユーザにさせるための方法を提供するものである。これらの技術は、また、メイン・ネットワークから切り離されたプロセス制御システムの一部を設定するか、又は同一の設定機器から複数のプロセス制御システムを設定するための機構をユーザに提供するものである。これらの技術は、また、複数のユーザに、リモートでシステムの異なる部分のコンフィギュレーションについて作業することを可能にし、プロセス制御システムの開発の全体的な並列性を改善することができる。

【0029】プロセス制御システムに用いられるブリーフケース・データベースの概念について、図3〜図5に基づいて説明する。概括して言えば、この構成を用いるプロセス制御システムは、メイン・サイトおよび異なるリモート・サイトを含む、プロセス制御システムの全てのサイトのために、コンフィギュレーション・データの全てを格納するマスター・コンフィギュレーション・データベースを備えるようになされてある。前記メイン・サイトおよびリモート・サイトは、衛星又はセルラー回線のような遅い又は小さい帯域幅の通信回線を介して通信可能に接続されても、モデム又は電話回線を介して間欠的に接続されても、又は全く接続されなくてもよいが、コンフィギュレーション・データを1つのサイトか

ら別のサイトへ転送するために、携帯型コンピュータ又はその他の携帯型メモリに依存している。一般的に、メイン・サイトをリモート・サイトに接続する各リモート・サイト又はラップトップ若しくはその他のコンピュータは、リモート・サイトで使用されるべきマスター・コンフィギュレーション・データベース内の幾つか又は全てのコンフィギュレーション・データのコピーを格納するブリーフケース・データベースを備えており、メイン・サイトは、マスター・コンフィギュレーション・データベースを備えている。

【0030】前記ブリーフケース・コンフィギュレーション・データベースは、ダウンロードおよび格納操作を用いてユーザによってマスター・コンフィギュレーション・データベースから初期化される。ダウンロード操作において、前記ブリーフケース・データベースは、リモート・サイトで作業するために、また、リモート・サイトを再設定、変更、デバッグ等するために必要であるマスター・コンフィギュレーション・データベース内のデータの一部のダウンロードをコピー又は受信する。前記ダウンロード操作では、リモート・サイトの状態を見ること等、リモート・サイトでの要求操作を実行するためにユーザによって必要とされるコンフィギュレーション情報の全てを自動的にダウンロードすることができる。任意の操作のために必要とされるコンフィギュレーション・コンポーネントの全てのリストは、生成され、格納され、また、前記ダウンロードを実行するために使用されることが可能である。これに代えて、ユーザが、リモート・サイトでの動作に必要とされる重要なコンフィギュレーション・オブジェクトを選択するようにしてもよく、また、これらの選択されたコンポーネントの親および／又は子供の幾つか又は全部が、ブリーフケース・データベースに自動的にダウンロードされるようにしてもよい。望むならば、しかしながら、ユーザは、特定のコンポーネントをダウンロードすべく選択することも可能である。

【0031】このデータがブリーフケース・データベースによって使用されている一方、コンフィギュレーション・データがマスター・コンフィギュレーション・データベース内で変更されることを防ぐために予約操作を使用する。ブリーフケース・データベースによってマスター・コンフィギュレーション・データベース内で予約された項目だけは、リモート・サイトで変更することができる。項目を予約するために、コンフィギュレーション・アプリケーションは、マスター・コンフィギュレーション・データベースに予約メッセージを送信することが可能であり、そして、予約済であるとしてマスター・コンフィギュレーション・データベース内のこの項目に目印を付け、他のユーザがマスター・コンフィギュレーション・データベース内の該当する項目に変更を行うことから防ぐようにしてある。この項目の読み込みの防止を

予約したときには、前記項目を任意にロックすることが可能である。このようなロック状態は、他人がこの項目を編集又は予約することを防止する。勿論、要求に応じて、前記ダウンロードおよび予約操作を、遅いか、小さい帯域幅であるか、又は間欠的な回線を介して実行することも可能である。しかし、コンフィギュレーション・データの全てが一緒に又は同時にダウンロードされるので、このようなダウンロードは、必要に応じて、個々に基づく遅い又は小さい帯域幅の回線を介してマスター・コンフィギュレーション・データベースから前記データのコピーを得ることよりも効率的である。

【0032】コンフィギュレーション項目をダウンロードおよび予約した後に、ブリーフケース・データベースが、リモート・サイト（ブリーフケース・データベースが既にリモート・サイトにない場合）へ持っていかれて、コンフィギュレーション動作を実行するためにリモート・サイトでコンフィギュレーション・アプリケーションによって使用される。特定の操作を実行するのに必要なコンフィギュレーション情報の全てがブリーフケース・データベースに提供されたので、ブリーフケース・データベースが直接的にリモート・サイトで使用され得るか、リモート・サイトでローカル・データベースを構築するために使用され得る。そして、このローカル・データベースがリモート・サイトで実行される任意のコンフィギュレーション・アプリケーションによって使用され得る。

【0033】一般的に、ブリーフケース・データベースを使用するコンフィギュレーション・アプリケーションは、そのブリーフケース・データベースによってマスター・コンフィギュレーション・データベースから予約されている項目を編集することのみができる。例えば、ユーザが、接続された複合体（linked composite）でモジュールを予約したならば、マスター・コンフィギュレーション・データベースは、ブリーフケース・データベースへ、モジュールと、そのモジュールの従属物又は子供の全部とを転送し、前記モジュールを予約するだけである。勿論、前記接続された複合体を、モジュールとして同様に予約することもできる。ユーザがダウンロード能力を必要とするならば、システムは、また、ユーザに特定のノード上でのダウンロード権利を予約させ、ブリーフケース・データベースで、それらのノードのためのダウンロード記録と状態情報とを含むコンフィギュレーション情報を交換させることができる。望むならば、ブリーフケース・データベースを使用するコンフィギュレーション・アプリケーションは、ユーザが編集することができる項目を明らかに示すように、ブリーフケース・データベースに格納されるコンフィギュレーション・データを表示することが可能である。例えば、編集可能な項目は、ソリッド・タイプ（solid type）においてコンフィギュレーション階層ツリー上で確認されることが可能

であり、ユーザが変更することができない全ての項目は、灰色にされて表示され得る。勿論、どの項目又はコンポーネントが予約されているかについてのその他の方法もまた、同様に使用され得る。上述のように、予約されているマスター・コンフィギュレーション・データベースの項目は、マスター・コンフィギュレーション・データベースにおいてロックされており、それらの項目を予約したブリーフケース・データベースによって発生した起用操作 (promote operation、下記) 以外には、それに対しては全く変更がなされないようにすることが可能である。

【0034】ユーザがブリーフケース・データベース内で予約された項目に変更をした後で、ブリーフケース・データベースは、それから、起用操作を通じて、マスター・コンフィギュレーション・データベースと整合させることができ、予約されたコンフィギュレーション・コンポーネントになされる変更は、マスター・コンフィギュレーション・データベースへリリースされて戻される。特に、起用操作は、これらの変更が反映されるか又はマスター・コンフィギュレーション・データベースに格納されるマスター・コンフィギュレーション・データベースへ、該マスター・コンフィギュレーション・データベースから予約されているコンフィギュレーション項目を起用する。望むならば、このような操作は、マスター・コンフィギュレーション・データベース内の項目の予約解除をする。コンフィギュレーション・アプリケーションは、遅い通信回線でのロードを低減するために、起用されている項目への値又は変更と共にマスター・コンフィギュレーション・データベースに起用メッセージを一度に送信することが可能である。勿論、予約および起用する操作は、コンフィギュレーション・データの単一の項目 (例えば、ライブラリ項目、モジュール、ノード、カード等) についてなされることができる。又はツリーの全部のセクションについて (例えば、親の、および、その子供および孫の全部について) なされることができる。

【0035】望むならば、コンフィギュレーション・アプリケーションは、例えば、ブリーフケース・データベースから予約された項目をエクスポートする、当該エクスポート・ファイルをジップで圧縮 (zip) する、また、マスター・データベース内のノード (即ち、遅い又は小さい帯域幅の通信回線を介して) ヘジップで圧縮されたエクスポート・ファイルを転送する等によって、マスター・コンフィギュレーション・データベースにブリーフケース・データベースの全体と整合させることが可能である。マスター・データベースでは、アプリケーションは、エクスポート・ファイルをジップで解凍し、任意の公知の方法によってマスター・データベース内へこのファイルをインポートし、マスター・データベースにおいて使用されるテキスト又はデータ形式のフォーマッ

トへマスター・データベースのための現在の設定を取り出し、エクスポート・ファイルをジップで圧縮し、また、ブリーフケース・データベースを備えるノードヘジップで圧縮されたエクスポート・ファイルを転送し返すことが可能である。ブリーフケース・データベースでのアプリケーションは、それから、エクスポート・ファイルをジップで解凍して、このジップで解凍されたファイルをブリーフケース・データベースへインポートし戻す。この処理は、マスター・コンフィギュレーション・データベースおよびブリーフケース・コンフィギュレーション・データベース内の項目の全てが同一であることを保証する。

【0036】さて、図3には、上述の如くブリーフケース・データベースを使用するプロセス制御システム100のコンフィギュレーション動作を図示してある。プロセス制御システム100は、制御動作を行うコントローラ (プロセス制御システム) 10に接続されたユーザ・インタフェース102、104を有するメイン・サイト101を備えている。また、ユーザ・インタフェース102、104は、ランタイム・データ・サーバ (ランタイム・サーバ) 108に接続され、更に、遅い通信回線 (例えば、衛星回線、モデム回線 (接続)、セルラー回線等) を介して、リモート・サイト111と通信するコンフィギュレーション・データベース・サーバ (データベース・サーバ) 110に接続されている。また、メイン・サイト101は、データベース・サーバ110によってアクセス可能なマスター・コンフィギュレーション・データベース112を備えている。同様に、リモート・サイト111は、ランタイム・サーバ118に、また、前記遅い回線を介してメイン・サイト101と通信するコンフィギュレーション・データベース・サーバ120に接続されたユーザ・インタフェース114、116を備えている。また、リモート・サイト111は、ブリーフケース・コンフィギュレーション・データベース122を備えている。ブリーフケース・コンフィギュレーション・データベース122は、リモート・サイト111のためのローカル・データベースとして使用されることができ、メイン・サイト101でマスター・コンフィギュレーション・データベース112から遅い回線を通じてコンフィギュレーション情報を得ることが可能である。

【0037】ユーザ・インタフェース114、116のうちの1つからのリモート、閲覧、予約、および整合操作を提供するコンフィギュレーション・アプリケーションを使用することにより、エンジニアは、マスター・コンフィギュレーション・データベース112からブリーフケース・データベース122へコンフィギュレーション情報をダウンロードすることができる。この手順の一部として、ユーザは、マスター・データベース112をコピーし、マスター・データベース112のためのツリー構造をコピーし、マスター・データベース112から如何なる構造をもコピーすることなくローカル又はブリーフケース

・データベースを構築する機会を与えられる。この際に、マスター・コンフィギュレーション・データベース112内で、リモート・サイト111におけるユーザは、また、ダウンロードされた情報の幾つか又は全てを予約することができ、これによって、その情報が変更されること、また、他人がマスター・データベース112内のその情報を変更しようとするのを防止することができる。好ましくは、前記ダウンロード操作は、遅い通信回線を通じて直ぐに必要なコンフィギュレーション情報の全てを提供する。そして、それは遅い回線を通じて少しずつこのデータを得ることよりも効率的である。

【0038】その後で、ユーザは、コンフィギュレーション・アプリケーションを使用しているリモート・サイト111又はそのコンポーネントと、リモート・サイト111でのブリーフケース・データベース122とを設定することができる。望むならば、これに加えて又はこれに代えて、ユーザがリモート・サイト111からメイン・サイト101の一部を設定することもできる。この際に、例えばユーザ・インタフェース116において実行されるコンフィギュレーション・アプリケーションは、ブリーフケース・データベース122にアクセスすることが可能であり、ブリーフケース・データベース122内に格納されたコンフィギュレーション階層若しくはツリーの一部を表示又は使用することが可能である。ユーザがマスター・データベース112において構造を見たいならば、ユーザは、整合操作を実行させることができる。遅い回線を介してマスター・データベース112に、その中の情報の全てについて、ツリー構造について、又はマスター・データベース112内のコンフィギュレーション情報のその他の要求された部分について照会し、この情報でブリーフケース・データベース122を更新する。予約された項目についてのコンフィギュレーション動作が終了すると、ユーザは、変更がマスター・コンフィギュレーション・データベース112に格納されるようにするために、遅い回線を介してマスター・データベース112へ前記予約された項目になされた変更を起用することが可能である。

【0039】さて、図4においては、一又は複数のブリーフケース・データベースが、非接続のワークステーションからコンフィギュレーション動作を実行するために使用されることが可能である。特に、図4は、マスター・コンフィギュレーション・データベース（マスター・データベース）132、コンフィギュレーション・データベース・サーバ（データベース・サーバ）133、およびこれに関連付けられたユーザ・インタフェース134を備えたメイン・サイト131を含むプロセス制御システム130を示している。リモート・サイト135、136の各々は、データベース・サーバ138と、一又は複数のユーザ・インタフェースと、コンフィギュレーション・アプリケーションを実行することが可能であるワークステーション140と、マスター・コンフィギュレーション・データベー

ス132において任意の項目についてのコンフィギュレーション動作を実行することが可能である一又は複数のブリーフケース・コンフィギュレーション・データベース142とを同様に備えている。望むならば、ブリーフケース・データベース142は、ユーザ・インタフェース装置140のうちの1つ内に存在することが可能である。図3のシステムと同様に、ワークステーション140におけるユーザは、マスター・データベース132に接続するために、そこからコンフィギュレーション情報をダウンロードするために、そして、変更されるべき項目の全てについてマスター・コンフィギュレーション・データベース132内で予約するために、例えば、インタフェース140のうちの1つにおいて実行されるコンフィギュレーション・アプリケーションを使用することが可能である。このようなダウンロードは、例えば、ダイヤルアップ接続又は携帯型の取り外し可能な物理的な接続（例えば、ラップトップ・コンピュータ）のような、遅い回線を通じて又は間欠的な接続を介して実行され得る。何れにしても、異なるサイト135、136は、マスター・コンフィギュレーション・データベース132からブリーフケース・データベース142へ、同一のコンフィギュレーション情報の幾つか又は全てをダウンロードする。しかし、異なるリモート・サイト135、136におけるユーザは、変更すべき別々の項目を予約することが可能である。

【0040】ダウンロード手順が実行された後で、コンフィギュレーション・アプリケーションは、各ブリーフケース・データベース142内で予約されている項目を変更又は修正するために、ワークステーション140上で実行されることが可能であり、若干の点で、これらの予約された項目への変更は、起用手順を使用するマスター・コンフィギュレーション・データベース132内へアップロードし戻されることが可能である。このように、例えば、エンジニアが、どこか他で働くために項目をワークステーション140内へ引き入れ、これらの項目を変更し、これらの項目のより新しいバージョンをマスター・データベース132内へ後日起用するために、ブリーフケース・データベース142は、メイン・ネットワーク131から完全に非接続とされた又は遅い通信回線を介して接続されたワークステーション140上で使用されることが可能である。同時に構築されるべき多数の異なるブリーフケース・データベース142を可能にすることによって、コンカレント・エンジニアリングを向上することができる。特に、コンフィギュレーション操作が論理的に分けられ、また、異なるエンジニアによって割り当ておよび完了されるように、各ブリーフケース・データベース142は、マスター・コンフィギュレーション・データベース132内のコンフィギュレーションのサブセットを含むことが可能である。

【0041】望むならば、新しいコンフィギュレーション・データを、異なるブリーフケース・データベース14

2間でデータが流れるのを可能にするために、マスター・データベース132へ定期的に起用し戻すことができる。例えば、実際には、1人のエンジニアが1つの設備に取り組み、他の人がその設備を動作している制御モジュールに取り組むことが可能である。後者は、制御モジュールのコンフィギュレーション・コンポーネントを完了する前に、設備のコンフィギュレーション・コンポーネントの最終的なバージョンを必要とする。しかし、最終的な設備定義については、前者のエンジニアが、前記設備定義を完了し、マスター・データベース132内へこの定義の内容を起用した後で、マスター・データベース132から引き出すことができる。

【0042】図5は、ブリーフケース・データベースが有利に使用されることが可能であるプロセス制御システム145の別の例を示す。該プロセス制御システム145は、データベース・サーバ148、ユーザ・ワークステーション149、およびランタイム・サーバ150に接続されたマスター・コンフィギュレーション・データベース（マスター・データベース）147を有する制御室を備えている。また、前記システム145は、LAN接続（共有LANを含む任意の種類のLAN接続であることが可能である）により制御室に接続され、データベース・サーバ152およびランタイム・サーバ154を有するプラント・フロアを備えている。また、前記プラント・フロアは、これに接続された又はこれが備えるブリーフケース・データベース156（例えば、ラップトップ・コンピュータ158又はその他のコンピュータ、ユーザ・インタフェース、又は例えばプラント技術者又はエンジニアがプラント・フロアで変更を行うことによって使用されるワークステーション）を備えている。プラント・フロアにおけるユーザは、LANを介してマスター・データベース147に格納されたコンフィギュレーション・データの項目を得るために、また、これらの項目の少なくとも幾つかを予約するために、コンピュータ158からコンフィギュレーション・アプリケーションを実行させることができる。その後で、プラント・フロアでコンフィギュレーションの変更をするためにコンフィギュレーション・アプリケーションで必要とされるその他の任意の項目と共に、予約された項目が、ブリーフケース・データベース156に一旦ダウンロードされたときに、前記プラント・フロアにおけるユーザは、予約された項目に変更をするためにローカル又はブリーフケース・データベース156を使用してコンフィギュレーション・アプリケーションを実行させることができ、また、LANを介してマスター・データベース147へこれらの変更を起用することができる。この処理は、マスター・データベース147に共有LANを通じて多数の呼び出しをする必要性を排除するが、能率が悪い。

【0043】同時に、家庭におけるユーザは、データベース・サーバ164を介してマスター・データベース147に接続されたコンピュータ160およびブリーフケース・デ

ータベース162を備えることが可能であって、例えば、ダイヤルアップ・モデムであり得る。このユーザは、モデム164を介してマスター・データベース147からコンフィギュレーション・データを得ることができ、この情報をブリーフケース・データベース162に格納することができる。また、このユーザが変更を要望する項目を予約することができる。完了すると、家庭におけるユーザは、暇なときに、ブリーフケース・データベース162内でコンフィギュレーション情報を操作するコンフィギュレーション・アプリケーションを実行し、ダイヤルアップ・モデムを介してマスター・データベース147へ変更を起用することができる。

【0044】望むならば、例えば、ユーザは、2つの異なるプロセス制御システムに変更をしたいと要望すれば、同一のブリーフケース・データベースにおける異なるプロセス制御システム、サイト等に関連する情報をダウンロード又は配置することができる。このような状況においては、複数のブリーフケース・データベースは、1つのコンピュータに格納されることが可能であり、この1つのコンピュータが同じプロセスの異なるサイトを設定するか、異なるプロセスに関連する異なるサイトを設定するために使用され得る。一般的に、このようなコンピュータは、コンフィギュレーション・アプリケーションを格納および実行し、コンフィギュレーション・データベースを有するラップトップ又はその他の携帯型コンピュータであって、アプリケーションにロードおよびアクセスするデータベースを伴っている。

【0045】望むならば、図1および図3～図5の任意のシステムのマスター・データベースは、コンフィギュレーション項目への変更に対するソース制御を提供するために、そこにコンフィギュレーション項目の改訂履歴を自動的に格納することが可能である。同様に、ブリーフケースのバージョンが変更され、起用動作が実行される必要があることを示すために、修正されているブリーフケース・データベース内の項目は、それに付けられた視覚的なチェックマークを有するように表示されることが可能である。

【0046】ブリーフケース・データベースを提供する代わりに、ミラー・データベースは、プロセス制御システムのリモート・サイトに保持されることが可能である。前記ミラー・データベースは、2つのデータベース（即ち、整合されたリモート・データベースおよびマスター・データベース）を保持することによってブリーフケースのモデルを拡張するものであって、例えば、遅い回線を通じてデータベースを編集することが実現可能ではない陸上／沖合でのプロセス制御構成においては、望ましいことである。また、前記ミラー・データベースは、冗長性をサポートするために使用されることが可能である。ミラー・データベースにおいて、例えば、ライブラリ、セットアップ、および制御データを含む全ての

コンフィギュレーション項目は、マスター・データベースからミラー・データベースによって予約される。このような予約は、望むならば、ここで図7～図15に関連して記述される方法において実行されることが可能である。さらに、コンフィギュレーション項目がミラー・データベースにおいて編集される際、当該コンフィギュレーション項目は、マスター・データベースから自動的に予約される。このように、前記ミラー・データベースにおける編集は、その項目を予約するために、遅い通信回線を通じて、メッセージをマスター・コンフィギュレーション・データベースへ自動的に送信する。予約操作がマスター・コンフィギュレーション・データベース内のその項目のロックのために失敗した場合、前記ミラー・データベース内の編集は防止される。その後の適当な時間で、変更がミラー・データベースにおいてなされた後で、コンフィギュレーション項目は、ミラー・データベースからマスター・コンフィギュレーション・データベースへ起用し戻される。このような起用動作は、変更がなされるとき、定期的な時間単位で（例えば、10分毎に、又は起用動作を実行するためのユーザ・コマンドに応じて手動で）、自動的に達成されることが可能である。

【0047】逆に、マスター・データベースへのミラー・データベースによる予約のために、マスター・コンフィギュレーション・データベースへの変更は、直ちに強制されるか、又は前記遅い通信回線を通じてミラー・データベースに送信される。ミラー・データベースとともに、通信回線がダウンしているときには、両方のデータベースは独立して動作することができる。しかし、前記通信回線が回復されたときには、前記データベースへの変更は手動で統合されなければならない。一般に、複数のミラー・データベースは、同一のマスター・データベースの予約者として存在することができる。ミラー・データベースおよびマスター・データベース間の接続の予約特性のために、ミラー・データベースが一旦確立された場合、ミラー・データベースおよびマスター・データベース内のデータへの変更だけが、遅い又は小さい帯域幅の通信回線を通じて送信される必要がある。そして、それは、ミラー・データベースのユーザの各々によって必要に応じて少しずつコンフィギュレーション・データをダウンロードしようとするよりも効率的である。

【0048】マスター・コンフィギュレーション・データベースと、又はこのセットアップに加えて、定期的に整合させることが可能である単一のマスター・コンフィギュレーション・データベースおよび又は複数のブリーフケース又はミラー・データベースを有する代わりに、コンフィギュレーション・データの異なるコンポーネントがプロセス制御システムの異なる物理サイトに存在する異なる物理データベースに格納されるように、コンフィギュレーション・データベースは、プロセス制御

システムの異なる部分に亘って分散されることが可能である。この分散型のコンフィギュレーション・データベースが、2つ以上の地理的に離隔したサイト又は位置に関連するコンフィギュレーション情報を一つに統合することに使用されることが可能であり、また、プロセス制御システム全体の継ぎ目の無い表示を提供することが可能である。ここで、この継ぎ目の無い表示とは、異なるサイト又は位置の各々におけるプロセス制御コンフィギュレーションのコンポーネントの幾つか又は全部を含むことが可能である。

【0049】図6は、異なる論理的なおよび／又は物理的な位置に関連付けられたコンフィギュレーション・データベースの階層を有するプロセス制御システム170を示す。該プロセス制御システム170は、Zone\_A 172, Zone\_B 174, およびZone\_C 176と名付けられた3つのゾーンと、Site\_1 180およびSite\_2 182と名付けられた2つのサイトと、企業ネットワーク（企業システム）184とを備えている。ゾーン172, 174は、コンフィギュレーション・データベース172a, 174aをそれぞれ備えており、これらのゾーン172, 174は、例えば、衛星、モデム、その他の遅く小さい帯域幅であるか、又は間欠的な通信回線を介して、サイト180に通信可能に接続されている（ゾーン172, 174とサイト180との間で実線により示す）。同様に、ゾーン176は、コンフィギュレーション・データベース176aを備え、任意の好ましい通信回線を介してサイト182に通信可能に接続されている。さらに、サイト180, 182は、コンフィギュレーション・データベース180a, 182aをそれぞれ備え、任意の好ましい通信回線を介して企業システム184に接続されている。また、該企業システム184は、コンフィギュレーション・データベース184aを備えている。

【0050】ゾーン172, 174, 176の何れかのように、ゾーンは、概して論理的であり、多くの場合に大きい制御システムの物理的な区域であることが可能である。このように、ゾーンは、一般に外界との繋がり無しで機能することができる。ゾーンは、例えば、特定の地理的な位置での相互接続されたデバイスを有する任意の従来のプロセス制御システムであることが可能である。サイト（サイト180, 182の何れか）は、エリア、範囲等の論理的な定義であって、これとともに関連付けられる任意の数のゾーンを有することが可能である。企業システム184は、プロセス制御システム170内のコンフィギュレーション・データベースの階層内の最も高水準のシステムであって、前記サイトの各々に、また、それによって、プロセス制御システム170内のゾーンの各々に通信可能に接続されている。勿論、その他のサイト、ゾーン、およびエリア、又はその他の論理的な単位は、遅い又はその他の通信回線を介して、プロセス制御システム170の階層に相互接続されることができる。そして、任意のゾーン、サイト、エリア等のコンフィギュレーション・デー

データベースがその他の任意のゾーン、サイト、エリア等から、2つ以上の通信回線の一又は一連のものを介してアクセスできる限り、プロセス制御システムの異なる地理的な位置は、その他の方法で相互接続されることが可能である。また、図6の分散型のコンフィギュレーション・データベースの階層が3つのレベル（ゾーン、サイト、および企業）を含んでいるが、2つのレベル又は4つ以上のレベルを代わりに使用することもできる。

【0051】図6に示す如く、プロセス制御システム170の各部分は、これに関連付けられたコンフィギュレーション・データベースを有し、プロセス制御システム170全体のためのコンフィギュレーション情報又はその異なる要素（コンポーネント）は、データベース172a, 174a, 176a, 180a, 182a, 184aを介して分散される。好ましい実施形態において、前記コンフィギュレーション・データ又はコンフィギュレーション・コンポーネントは、そのデータがユニークな名前を付けられている企業、サイト、およびゾーンによって形成される階層において最も低いレベルで、コンフィギュレーション・データベースに格納される。このように、例えば、プロセス制御システム170において、何処にでも使用され得るデバイスとソフトウェア・コンポーネントとのテンプレートを格納することが可能であるライブラリ・データは、概して、階層において（例えば、企業システム184のコンフィギュレーション・データベース184aにおいて）最も高いレベルに格納され、また、最も高いレベルからアクセスできる。同様に、ライブラリ・コンフィギュレーション・コンポーネントとサイトのシステム・コンフィギュレーションとに関連する情報は、ゾーンにおける特定のデバイス、制御モジュール等に関連するコンフィギュレーション情報が、そのゾーンのコンフィギュレーション・データベースに格納される一方、前記サイトのコンフィギュレーション・データベースに格納されることが可能である。さらに、このコンフィギュレーション情報がそのゾーン内の物理的なデバイスに直接関連があるので、制御のコンフィギュレーション情報は、ゾーン・コンフィギュレーション・データベースに概して格納される。概括して言えば、目標は、このコンフィギュレーション・データが最もアクセスされそうであるように、又はコンフィギュレーション・データの任意の特定の部分がプロセス制御システム170のその他の位置におけるユーザによって容易に見つけられるように、データベースの階層のコンフィギュレーション・データベース内の任意の特定の部分を格納することにある。

【0052】図6に示した分散型のコンフィギュレーション・データベースの方法によれば、プロセス制御システム170のゾーン、サイト、エリア等のうちの任意の1つ内で動作するコンフィギュレーション・アプリケーションは、同一のサイト、ゾーン等、および／又はその他のサイト、ゾーン等に配置されることが可能である一又は

複数のコンフィギュレーション・データベースから、コンフィギュレーション・データにアクセスし、また、プロセス制御システム170の現在の状態を示すために、又はその他の何らかのコンフィギュレーション操作を実行するために必要であるとして、そのデータにアクセスする。このように、図6のゾーン、サイト、又は企業システムのうちの何れから実行されるコンフィギュレーション・アプリケーションは、プロセス制御システム170の任意の部分又は全部の現在のコンフィギュレーションの表示を生成するために、異なるコンフィギュレーション・データベースの何れかに格納されたコンフィギュレーション・データにアクセスすることができ、また、プロセス制御システム170のサイト、ゾーン等の何れかの要素（コンポーネント）の何れかを再設定可能にすることもできる。分散型のコンフィギュレーション・データベース（例えば、図6に示したもの）は、遅い通信回線を通じて発生する必要がある通信量を減少させる傾向がある。それは、概して言って、サイト又はゾーンにおけるユーザは、異なるサイト又はゾーンのそれよりもむしろそのサイト又はゾーンのコンフィギュレーション・データを見て、使用して、また、変更しそうであるからである。即ち、より少ない読み書きが一又は複数の遅い通信回線を介してその他のゾーン又はサイトからやって来るが、サイト又はゾーン内のコンフィギュレーション・データベースへの読み書きの殆どは、そのサイト又はゾーン内のユーザ・インタフェースからやって来るのである。しかし、プロセス制御システム170の任意の部分のコンフィギュレーション情報の全てに対しては、プロセス制御システム170のその他の任意の部分におけるユーザ・インタフェース又はその他のデバイスが依然として利用可能である。

【0053】異なるゾーン、サイト等の間での通信を可能にするために、前記システム内の各データベースは、コンフィギュレーション階層の一般的なセットアップに関連する情報を格納することが可能である。例えば、各データベースは、全ての位置でのコンフィギュレーションに共通である幾つかのオブジェクト・ルーツ（object root）を格納することが可能であり、これらのルーツに関する更なるコンフィギュレーション情報を有する特定のデータベースを示す、又はこの情報を求めるために次のデータベースを示す、ポインタを格納することが可能である。このように、Zone\_A 172におけるユーザは、ルーツ・オブジェクトにアクセスし、そのルーツの子供を要求しようとするのが可能である。これに応じて、コンフィギュレーション・データベース172aは、選択されたルーツのためのポインタを見ることによって、子供が格納されているデータベースを見つけることが可能である。そして、このポインタは、そのルーツのためのデータが企業システム・データベース184aからアクセスされなければならない、又は「Site\_1」データベース180a



がそのデータへ更なるポインタのためにアクセスされなければならないことを示すことが可能である。データベース172aは、遅い回線を介したデータベース180aの読み出しをさせ、データベース180aに企業システム・データベース184aの読み出しを実行させることが可能である。勿論、企業システム・データベース184aは、「Site\_2」データベース182aにアクセスする必要がでてくる可能性があり。そのデータのために「Zone\_C」データベース176aにアクセスする必要がでてくる可能性がある。代わりに、ブラウザ又はスニッフ・アプリケーション (sniffing application) は、例えば、「Zone\_A」データベース172aを、分散型のコンフィギュレーション・システム170内のコンフィギュレーション・データの任意の特定の部分の位置を見つけるために「Site\_1」データベース又はその他のデータベースを閲覧することが可能なようにするために、各データベースに提供されることが可能である。

【0054】望むならば、システム170内の各データベースは、元々異なるサイト、ゾーン等に格納されたデータのローカル・コピーを格納することが可能である。特に、特定のデータベースに格納されないデータに対する要求が受信された場合、該データベースは、そのデータを異なるデータベースから引き出すことが可能であり、このデータを取り戻すことに際して、直接、データベース (例えば、同じサイト又はゾーンにおけるユーザ) に接続されたユーザによって使用されるように、又はデータベースの階層内の他のデータベースによる要求に応じて送信されるように、それをローカルに格納することが可能である。このように、上の例では、データベース180a, 184a, 182aの各々は、「Zone\_C」データベース176aに元々格納されたコンフィギュレーション・データのローカル・コピーを格納することが可能であり、これは、そのデータがZone\_Aで「Zone\_A」データベース172a又はユーザによって要求されたからである。

【0055】任意のユーザ・インタフェースで表示された特定のコンフィギュレーション表示のためのコンフィギュレーション・データの幾つか又は全部が、一又は複数の遅い通信回線を介してアクセスされなければならないかも知れないので、同一のサイト若しくはゾーン又は異なるものの何れかにおいて、別のユーザによってコンフィギュレーション情報になされた変更を基準として、特定のユーザ・インタフェースで表示されたコンフィギュレーション・データが新鮮ではないか又は古くなる状況に出くわすことがある。各コンフィギュレーション・アプリケーションが画面上に提示されているコンフィギュレーション情報をリフレッシュ (refresh) するために適当なコンフィギュレーション・データベースに定期的にアクセスすることができる。この周期的なリフレッシュは、遅い通信回線を通じて発生する筈の、また、コンフィギュレーション・データベースからの読み出しを

増加する、通信量を実質的に増大し、プロセス制御システム170内の通信を遅くし、そのコンフィギュレーション・データベースの各々へのロードを増加する。

【0056】この問題を解決するために、コンフィギュレーション・データにアクセスしている各コンフィギュレーション・アプリケーションは、画面上に表示されている (又は使用されている) 各コンフィギュレーション・コンポーネントのために、適当なコンフィギュレーション・データベースを予約することが可能である。そして、このデータを格納するコンフィギュレーション・データベースは、コンフィギュレーション・データベース内で前記情報になされた如何なる変更をも各予約のコンフィギュレーション・アプリケーション (クライアント) に自動的に通知する。それから、これらの変更は、例えばユーザ画面上に表示されるべきクライアントに自動的に送信されることが可能であるか、又はユーザによって要求されることが可能である。このように、たとえ何処にこの情報を使用するクライアント・アプリケーションがプロセス制御システム170内で位置しようとも、各コンフィギュレーション・アプリケーション (例えば、各画面上に示されている情報の各部分) によって使用されているコンフィギュレーション情報は最新である。

【0057】勿論、クライアント・アプリケーションがコンフィギュレーション情報の特定の部分をもはや必要としないとき、即ち、例えば、ユーザがもはやコンフィギュレーション階層におけるコンフィギュレーション情報の特定の部分を見たくないときには、コンフィギュレーション・アプリケーションは、変更通知および更新が予約中のコンフィギュレーション・アプリケーションにもはや送り返されないように、その情報の部分を格納するコンフィギュレーション・データベースから予約解除する。このように、コンフィギュレーション・アプリケーションは、例えば、図2のツリー65内におけるコンフィギュレーション情報の各部分を格納するコンフィギュレーション・データベースを見つけ、必要に応じて一又は複数の遅い通信回線を通じてその情報を取り出し、この情報が格納されるコンフィギュレーション・データベース内における該情報になされた更新を受信するためにその中の情報の各部分を予約し、受信に際しては、表示された情報 (見られているシステムのコンフィギュレーションへの変更をしている別のユーザによるものであることが可能) を更新し、新しい又は更新された情報を表示する。ユーザ又はオペレータが、表示されているコンフィギュレーション・ツリーのセクションを掘り下げていくならば、コンフィギュレーション・アプリケーションは、同様に、新しい情報にアクセスし、この情報を予約し、そして、この情報を表示する。表示された情報の幾つかがユーザ画面から落とされるならば、コンフィギュレーション・アプリケーションは、落とされたデータ

からの予約を解除する。

【0058】望むならば、得られる（例えば、特定のユーザ・インタフェースの画面上で見られる）コンフィギュレーション情報の全ては、ローカル・キャッシュ又は実行されているコンフィギュレーション・アプリケーションに関連したメモリに格納されることが可能であるか、前記アプリケーションが実行されているゾーン、サイト等内のコンフィギュレーション・データベースに格納されることが可能である。そして、適当なデータベースとの接続を失った場合には、そのサイト、ゾーン等におけるユーザは、ローカル・キャッシュからこのコンフィギュレーション情報をまだ引き上げることが可能である。しかし、ユーザがこの情報が最新ではないことを知るように、この情報は、ローカル・キャッシュ（例えば、灰色表示のような）から来たこの情報を示すように表示されることが可能である。一又は複数の遅い回線を介して既に得られたコンフィギュレーション・データを格納するためのこのようなローカル・キャッシュを使用することは、通信線又は回線がダウンした場合に、ユーザに対して、該ユーザによって既に見られるた又はアクセスされた任意のコンフィギュレーション情報を見ることを可能にする。また、ユーザがその画面から落とされた情報を見たいならば、この情報は、該情報のために適当なコンフィギュレーション・データベースとの接続がその情報を更新するために再確立されている間、それをより速く表示させるためにローカル・キャッシュから引き出されることが可能である。

【0059】図6が遅い回線を介して相互接続されるようなサイトおよびゾーンの各々を示しているが、このような構成は必須ではない。例えば、サイトのためのコンフィギュレーション・データベースは、ゾーンにあるかも知れず、遅い通信回線を介してそのサイトに関連したその他のゾーンへアクセスできれば、そのゾーンから直接アクセスできるかも知れない。例えば、図6で図示されるような分散型のコンフィギュレーション・データベースを確立するために、Zone\_A 172におけるユーザは、「Zone\_A」データベース172aと同じマシン上で又は直接の高速接続を通じて「Zone\_A」データベース172aに接続している異なるマシン上でサイト・データベース（即ち、「Site\_1」データベース180a）を構築することが可能である。代わりに、「Site\_1」データベース180aは、遅い回線を介して「Zone\_A」データベース172aに接続していることが可能である。ユーザは、「Site\_1」データベース180aにおよび／又は企業システム・データベース184aにZone\_Aでライブラリおよびセットアップデータの幾つか又は全てを公開する。その後で、Zone\_A 172でのデータベース172aは、ユーザ又はZone\_A 172内のアプリケーションがそれらの項目を必要とするか、又はそれらの項目にアクセスしたいときに、「Site\_1」データベース180a、企業システム・データベース184a等内でコンフ

ィギュレーション項目のうちの1つ以上に自動的に予約することが可能である。望むならば、システム170の如何なるコンフィギュレーション・データベースであっても、そのシステムのその他のコンフィギュレーション・データベースのうちの1つ以上内で、コンフィギュレーション情報の幾つか又は全てに常を予約することができ

る。【0060】次に、Zone\_B 174におけるユーザは、Zone\_B 174のためのサイト・データベースとしてZone\_Aにおけるユーザによって構築される「Site\_1」データベース180aを選択することが可能である。それから、同一の名前の項目が「Site\_1」データベース180aおよび「Zone\_B」データベース174a内に存在するような衝突に終わり得る「Site\_1」データベース180aにおいて、Zone\_B 174におけるユーザは、ライブラリおよびセットアップデータの幾つか又は全てを予約する。コンフィギュレーション・アプリケーションは、Zone\_B 174におけるユーザへこの問題について通知することが可能であり、Zone\_B 174のユーザは、「Site\_1」データベース180aにおける項目を予約する前に、ローカルの項目の名前を変えるオプションを与えられることが可能であるか、ローカルの項目に上書きすることが可能である。加えて、Zone\_B 174におけるユーザは、「Site\_1」データベース180aに項目を公開することが可能であり、そして、その後で、Zone\_A 172は、「Zone\_A」が予約する「Site\_1」データベース180aへの追加を知らせられ、それから、これらの追加のコンフィギュレーション項目を予約することが可能である。勿論、企業／サイト階層は、サイト180、182の間と、ゾーン172、174、176の間とでの通信を可能にするために、同様の方法で設定される。

【0061】Zone\_A 172およびZone\_C 176間の接続を確立するために、例えば、Zone\_A 172におけるユーザがZone\_C 176のデータベース176aに格納されたコンフィギュレーション・コンポーネントを見たいとき、この情報を予約し、Site\_2 182からこの情報を予約する企業システム・データベース184aにこの情報を公開する「Site\_2」データベース182に、「Zone\_C」データベース176aはこのデータを公開する。Zone\_A 172が「Site\_1」データベース180からこの情報を予約する一方、Site\_1 180は、企業システム・データベース184aからこの情報を予約する。確立された予約者関係は、それから、「Zone\_C」データベース176aに格納されたコンフィギュレーション・コンポーネントへの変更について、Zone\_A 172におけるクライアント又はユーザに通知するために使用されることが可能である。上述のように、ゾーン、サイト等は、項目について個々に、又は親および関連する子供の全てを含むオブジェクトのサブ・ツリー全体を予約することができる。また、予約は、これらの項目をローカル・ゾーンのコンフィギュレーション・データベースに加え、公開された項目とローカル・コピーとの間の進行中の通

信関係を確立して、項目のコピーをとることを課すことが可能である。本来の情報がどこに格納されたかについて、予約中のデータベースがトラッキングする一方、公開者のデータベースは、予約者についてトラッキングすることが可能である。このソース／デスティネーション (source/destination) 情報が、コンフィギュレーション・データベース内の項目の予約を指示して、リモート位置から変更をなすために使用され得る。同様に、データベースがバックアップから回復しなければならないならば、この情報は、公開／予約関係を再確立するために使用されることが可能である。

【0062】勿論、ゾーン間の通信は、該ゾーン間の遅い通信回線の存在のために必ずしも保証されることができないので、通信回線がダウンしているときには、ゾーンは独立して機能し続ける。しかし、通信が再確立されるとき、如何なる衝突であっても手動で解決されなければならない。

【0063】ゾーン、サイト、および企業間の通信が、公開又は予約単位で実行されると言える。ここで、一又は複数の予約者がその項目を予約するときに、コンフィギュレーション項目のマスター・コピーを有するコンフィギュレーション・データベースが、その項目を公開する。予約者は、例えば、異なるサイト又はゾーン内のユーザ・インタフェース (例えば、ワークステーション又はその他のコンピュータ) において実行されるコンフィギュレーション・アプリケーションであることが可能であり、これらのサイト又はゾーン内のデータベース (それらのゾーンにおいて実行されるコンフィギュレーション・アプリケーションによって使用される) であることが可能であり、又はコンフィギュレーション・データを使用するその他の任意のデバイス又はアプリケーションであることが可能である。その他のサイト又はゾーンによって予約され得るデータベース内の項目は、共有オブジェクトと呼ばれる。勿論、次のような異なる種類の操作が共有オブジェクトについて行われることが可能である。1) 閲覧：予約者のコンフィギュレーション・アプリケーションが、公開者から入手可能な共有オブジェクトのリストを見ること。2) 予約：予約者のコンフィギュレーション・アプリケーションが、予約者の位置でのコンフィギュレーション動作に使用される前記共有オブジェクトのローカル・コピーを要求すること。3) ローカル・オブジェクトへのコピー：予約者のコンフィギュレーション・アプリケーションが、項目を非共有ローカル・オブジェクト又はデータベースへコピーすること。4) 削除：予約者のコンフィギュレーション・アプリケーションが、公開者において共有例を削除すること。5) 予約解除：予約者のコンフィギュレーション・アプリケーションが、予約者でのコンフィギュレーション項目のローカル・コピーと、公開者のコピーとの間の予約リンクを切断すること。6) 変更の取り出し：予約

者のコンフィギュレーション・アプリケーションが、公開者から共有項目の最新のバージョンを取り出すこと。

7) 予約：予約者のコンフィギュレーション・アプリケーションが、このオブジェクトが別のユーザによって変更されるのを防止するために、共有オブジェクトを公開者のデータベースにロックすること。8) 起用：予約者のコンフィギュレーション・アプリケーションが、予約された項目になされた変更を公開者のデータベースに戻すこと。これらの操作は、予約者と公開者との間でメッセージを送信することによって、また、メッセージを扱い、上記定義のような適切な動作をとるために、予約者の位置と公開者の位置とでソフトウェアを実行することによって行われ得る。

【0064】コンフィギュレーションの観点からは、セットアップおよびライブラリ・データを共有すること、別のゾーンからセットアップ、ライブラリ、およびコンフィギュレーション・データを閲覧又は見ること、ゾーン間において、ゾーン間関係およびネーム空間調整を設定することを含む、幾つかのゾーン間動作がある。1つの実施形態において、コンフィギュレーション・オブジェクト又は項目は、階層内の直近の親から共有される。例えば、ゾーン172, 174は、サイト180から合成定義 (composite definition) を共有し、全てのゾーンは、企業システム184から設定される共通の一覧表を共有する。ゾーン176がサイト182で定義を予約する一方、ゾーン172, 174は、サイト180で定義を予約する。同様に、両サイト180, 182は、企業システム184において共通の定義を予約する。

【0065】閲覧サービスは、企業／サイト／ゾーン階層においてサーバを見つけるために、また、それらの位置のうちの1つに関連付けられたコンフィギュレーション・データベースを閲覧するために、コンフィギュレーション・アプリケーション内に提供される。閲覧サービスは、共有オブジェクトへの予約をセットアップし、ゾーン間リファレンスを設定するために利用される。特に、閲覧情報は、ユーザ要求に応じてリモート・サーバから照会され、ローカルにキャッシュされる。閲覧ルート (browse root) は、ネットワークをスニフing (sniffing) することによって検出可能であるか、又はその他の任意の好ましい若しくは公知の閲覧技術を利用して検出可能である。検出された又は設定された閲覧ルートについての知識は、持続的であってもよく、電源スイッチを切って直ぐに入れなおすこと (power cycle) を通じてキャッシュされたままである。これらのコンフィギュレーション・オブジェクトのルートの存在は、ユーザに対して、分散型のコンフィギュレーション階層内のこの情報の位置又はこの情報への経路を見つけるために、前記ルートに関連付けられたその他の情報を閲覧することを可能にする。

【0066】ゾーン間リファレンスは、直接的にリファ

レンスを入力することによって、又はリモート・データベースにおける属性を閲覧することによって、設定されることが可能である。リファレンスは、例えば、「//ZoneName/TaggedItem/.../AttributeName」のような形式又はその他の任意の好ましい形式をとることが可能である。ゾーン名は、ゾーン間リファレンスを解決するために企業階層内でユニークでなければならないと言える。各ゾーン内には、タグを付けられた項目のためのネーム空間があり、コンフィギュレーション・システムは、ゾーン内でタグのユニーク性を強要され得る。このように、各コンフィギュレーション・アプリケーションは、分散型のコンフィギュレーション・データベース内の名前付けにおいて、ユニーク性を強要されるのに合わせて、タグがサイトか企業内でユニークかどうかについて判断するツールを有することが可能である。

【0067】望むならば、ゾーン間の安全は、任意の好ましい方法で提供されることが可能であり、ゾーン管理者は、ユーザ、グループ等のための共有オブジェクトへアクセスする権限が与えられ得る。このような権限は、閲覧して共有オブジェクトを予約する能力と、オブジェクト又はコンフィギュレーション項目のための変更を公開者から引き出す能力と、変更を予約、修正、および起用する能力とを含むことが可能である。

【0068】また、望むならば、異なるゾーンは、異なる言語を使用することが可能である。これは、会社が多くの異なる言語を使用している地域、例えば、ヨーロッパにおいて有益である。この場合、一覧化された値は、数値として、又は異なる言語で同一の単語又はフレーズのためにプロセス制御システムを通じて共通である数値とともに表示されることが可能であるローカル・データの文字列（例えば、単語）として、ゾーン間を通過することが可能である。このように、ユーザは、1つの言語から他の言語への変換の容易性を可能にする、それに応じた数値によって、一覧化された値、コマンド等を特定することが可能である。何れにしても、異なる言語を使用するデータベースを閲覧するためには、関連するフォントが閲覧マシン又はゾーン上にインストールされることが必要である。1つの実施形態において、ゾーン間リファレンスは、リモート・ゾーンの言語で設定される。そして、ユニコード・ストリング（Unicode）は、ユニコードによってサポートされた任意のキャラクタについての検出又は変換を可能にするために排他的にシステムを通じて使用される。この場合、単一のデータベースは、複数の言語のストリングを格納することが可能であり、データベースのエクスポート・ファイルは、ユニコード・フォーマットに交換される。言い換えれば、異なるローカルからのファイルは、ユニコード・フォーマットを使用して各データベースにインポートされる。

【0069】修正が干渉無しに実行され得ることを保証するために、共有データへの修正が上述した予約／起用

手順の対象となることが可能である。この場合、項目が修正され得る前に、修正するユーザは、該ユーザのゾーンへこの項目を予約しなければならない。1人のユーザ又はゾーンだけが、任意の特定の時間にでも項目を予約することができる。項目が予約された後で、この項目を予約したローカル・データベースにおいて、変更が当該項目になされる。そして、この項目を予約したユーザによって、このような変更が新しいコンフィギュレーション動作の結果としてなされる。その後で、予約され、修正された項目又は項目のグループは、このデータがデータのマスターセットとして格納されているコンフィギュレーション・データベースへ起用される。このような予約／起用手順は、それに2人のユーザが同時に、同一の項目を変更しようとすることができないことを保証する。その代わりに、項目を予約しようとする第2のユーザは、その項目を予約する第1のユーザが該項目のマスター・バージョンを格納するコンフィギュレーション・データベースへこの項目の変更を起用するまで、如何なる修正であっても、なすことも防止される。

【0070】通常、予約については2つのモードが使用される。最初のモード（ここでは、「安全予約モード」と称す）において、既に存在する項目への修正および該項目の削除が手動で引き入れられる一方、新しい共有項目は、予約しているゾーン又はサイトに自動的に引き入れられる（例えば、ユーザからの直接コマンドが前記変更又は削除情報の引き入れを実行するのに要求される）。この場合、ユーザは、ユーザ・システムが予約しているコンフィギュレーションの変更又は削除について通知されることが可能であり、また、コンフィギュレーション・アプリケーションがその変更をローカル・データベースへ引っ張っていくことになっているかどうかを尋ねられることが可能である。代わりに、共有データベースへの如何なる変更に対しても、ユーザの干渉無しで予約中のデータベースにインポートされるように、全ての修正は、自動的に引かれる（pull）ことができる。安全予約モードには、変更通知が全階層を通じて移行しない点に留意する必要がある。このように、例えば、企業の項目が修正されれば、依存しているサイトに通知される。しかし、その変更がサイトに実際に引き下げられるまで、そのサイトに関連付けられたゾーンがその変更について通知されることはない。その結果、公開者（即ち、そのコンフィギュレーション項目のマスター・コピーを格納するデータベース）は、予約者が項目の最新バージョンをとったかどうかについてトラッキングすることが可能である。そして、どちらかのデータベースをバックアップから回復しなければならないならば、この情報は、公開者／予約者関係を再整合するために使用されることが可能である。

【0071】さらに、変更を引く予約者の失敗が、コンフィギュレーション・データベースの階層における下位

のレベルでのコンフィギュレーション階層の違いにつながると言える。例えば、企業システム184が図6のサイト180、182に変更について通知し、Site\_1 180がこの変更を引かないで、Site\_2 182がこの変更を引くならば、サイト180、182で見られるコンフィギュレーション階層は異なる。さらに、変更を引いているSite\_2 182について、この変更を引くことができるか、又は変更が発生したということを少なくとも知ることができるZone\_C 176における予約者に変更通知が送信される。しかし、Site\_1 180が変更を引かなかったので、ZoneA 172とZone\_B 174とにおける予約者は変更について知らず、従って、Zone\_A 172とZone\_B 174とは変更を引くことができない。この場合、ゾーン172、174で利用できるコンフィギュレーション階層は、ゾーン176で利用できるものとは異なっている。望むならば、ユーザが変更に関して何が起きているかについてわかるのを手伝えるために、上流および下流の従属リストを表示するコンフィギュレーション・アプリケーションを提供してもよい。ユーザが項目を引こうとするとき、コンフィギュレーション・アプリケーションは、如何なる上流の改訂の不一致でも見つけることができ、また、これらの項目が引かれなければならないか否かをユーザに尋ねることができる。

【0072】さて、図7～図15を参照して、プロセス制御システム内のデータベースから効率的にアクセスおよび分散するためのシステムについて説明する。概括して言えば、データベース・アクセス・システムは、そのデータの一又は複数の予約者（クライアントとも称される）へ、データベースに格納されたデータを公開するために、また、データへの変更がデータベース内で発生するときに、予約者又はクライアントへ更新又は変更通知を提供するために共有キャッシュを使用する。ここに記述されているデータベース・アクセス・システムは、プロセス制御システム（例えば、図1のコンフィギュレーション・データベース30、54のうちの何れか、図3～図5のマスター・コンフィギュレーション・データベースの何れか、又は図6のコンフィギュレーション・データベースの何れか）内で、データベースの何れかに使用されることが可能であると言える。同様に、アクセスされているデータベース内のデータの予約者又はクライアントは、その他の任意のコンフィギュレーション・データベースであることができ、また、システム内の任意のユーザ・インタフェース又はコンフィギュレーション・データを使用する任意のコンフィギュレーション・アプリケーションにおいて実行される任意のアプリケーションであることができる。そして、これらのクライアントは、例えば直接又は高い速度の通信回線、遅い又は小さい帯域幅の通信回線、又は間欠的な通信回線を使用してアクセスされるデータベースに通信可能に接続されることが可能である。このように、例えば、図7～図15において記述されるクライアントは、図1および図3～図

6のゾーン又はサイトの何れかに配置され得る。

【0073】概括して言えば、各クライアントは、該クライアントが予約するコンフィギュレーション・コンポーネントを格納するコンフィギュレーション・データベースに関連するデータベース・サーバと通信する。前記予約プロセスの起動において、コンフィギュレーション・データベースに接続する各々の異なるクライアントのために、クライアント・スレッドが、データベース・サーバ内で確立される。そして、このスレッドは、コンフィギュレーション・データベース内で一又は複数の特定のコンポーネントへのアクセスを提供する。その後で、コンフィギュレーション・データベース内のコンポーネントへの変更が認識され、これらの変更の通知と、望むならば、この変更自体とが、そのコンフィギュレーション・コンポーネントを予約している各々のクライアントに送り返される。コンフィギュレーション・コンポーネントからクライアントが予約解除するならば、変更通知は、そのコンポーネントのためにクライアントへ提供されない。また、クライアントは、ロックし、予約し、起用するクライアント・スレッドを使用することが可能である一方、コンポーネントに変更をなすことが可能である。

【0074】用語「スレッド（thread）」は、データベース・サーバ内のプロセッサによって実行される処理パス又は処理手順を参照するものであり、並列処理を実行する公知の技術であると言える。概括して言えば、プロセッサは、実行されている異なるスレッドに関連するタスクをインターリーブ（interleaving）することによって、同時に複数のスレッドを実行することが可能である。例えば、前記プロセッサは、順に、第1のスレッドに関連付ける第1のステップ、第2のスレッドに関連付ける第1のステップ、第3のスレッドに関連付ける第1のステップ、それから、第1のスレッドに関連付けられた第2のステップは、第2のスレッドに関連付けられた第2のステップ、第3のスレッドに関連付けられた第2のステップ等を実行することが可能である。スレッドは、実行中の他のスレッドに影響を与えることなく、付加又は削除されることが可能である。スレッドは、スレッド間通知又はメッセージを使用している各々のその他のスレッドと通信することが可能である。また、格納され、異なるスレッドによって使用されるデータは、データを生成又は格納するスレッドによってアクセスされることが単に知られているか、又はアクセスされることができ、例えば、ローカルスレッドであり得る。

【0075】さて、図7において、複数のクライアントとコンフィギュレーション・データベースとの予約者関係を実現するコンフィギュレーション・データベースの通信システム200のブロック図は、コンフィギュレーション・データベース203と複数のクライアント206～208との間で通信可能に結合されたデータベース・サーバ20

2を含んでいる。コンフィギュレーション・データベース203は、2つのデータ・ストア210、212を備えるように図示しており、各々は、クライアント206~208のうちの又は複数によって予約されているコンフィギュレーション・コンポーネントを格納する。データベース203は、図示の明確化のために、2つのデータ・ストア210、212を有するように示してあるが、より多くの又はその他のデータ・ストアをその中に備えることができる。さらに、任意の数のクライアントがサーバ202を使用してデータベース203にアクセスすることが可能である。また、各データ・ストア210、212は、1つのサーバが1つ以上の異なるデータベースにアクセスするために使用され得るように、異なる物理データベースに関連付けられ得る。

【0076】クライアント206~208は、イーサネット回線のような、直接の、高い速度回線であることが可能である、又は、例えば、衛星回線、セルラー回線、モデム回線（接続）等のような、遅く小さい帯域幅の又は間欠的な回線であることが可能である通信回線213を介して、サーバ202と通信する。クライアント206~208によるコンフィギュレーション・コンポーネントのための各々の最初の要求は、サーバ202内にクライアント・スレッドを生成する。そして、以下に更に詳細に記述されるように、このクライアント・スレッドは、サーバ202内の共有キャッシュ214を使用している適切なデータ・ストア210および/又は212から、クライアント206~208に要求されたコンポーネントを提供するために使用される。また、クライアント・スレッドは、データ・ストア210、212に変更を書き込むために、データベース203へコンポーネント・データを加えるか又はデータベース203からコンポーネント・データを削除するために、また、データ・ストア210、212内のコンポーネント・データのロックを実行するために、使用され得る。

【0077】概括して言えば、共有キャッシュ214は、クライアント206~208の何れかによって任意の特定の時間に予約されているデータベース203内の異なるデータ項目の各々のための、「ライトウェイト (lightweight)」と呼ばれるデータ・メモリ・オブジェクトをセットアップする。1つのライトウェイトは、データベース203内の任意のコンフィギュレーション・コンポーネントのために確立されることを必要とする。そして、そのコンフィギュレーション・コンポーネントを予約している全てのクライアントのためのクライアント・スレッドは、同一のライトウェイトにアクセスする。このように、図7のシステムのために、クライアント206~208のうちの少なくとも1つがデータ・ストア210で前記コンポーネントを予約するときに、ライトウェイトは、データ・ストア210、212の中でコンフィギュレーション・コンポーネントの各々のために共有キャッシュ214内に確立される。そして、クライアント206~208のうちの少なく

とも1つは、データ・ストア212で前記コンポーネントを予約する。クライアント206~208の各々が異なるコンフィギュレーション・コンポーネントを予約するならば、各クライアントのためのスレッドは、コンフィギュレーション・データベース203内で異なるデータ・ストアに連結された異なるライトウェイトを含んでいる。しかし、1つ以上のクライアントがデータベース203内で同一のコンポーネントを予約するときには、これらのクライアントのためのスレッドは、共有キャッシュ214内で同一のライトウェイトを使用する。この理由のため、共有キャッシュ内のライトウェイトは、ローカルなスレッドではない。

【0078】通常、第1のクライアントがデータベース203内で最初にコンフィギュレーション・コンポーネントを予約するときにはいつでも、ライトウェイトは、共有キャッシュ214内でそのコンフィギュレーション・コンポーネントのために確立される。そして、このライトウェイトは、データベース203からコンフィギュレーション・コンポーネントのコピーを読み込み、格納する。その後で、同一のコンフィギュレーション・コンポーネントを予約するその他の全てのクライアントは、同一のライトウェイトに接続されたそのクライアント・スレッドを有する。これは、このコンフィギュレーション・コンポーネントに予約する第2、第3等のクライアントがコンフィギュレーション・データベース203からでなく、前記ライトウェイトからコンポーネントを読み込むことができる。つまり、コンフィギュレーション・データベース203からの読み込み回数を低減する。

【0079】図7に示すデータベース・サーバ202は、クライアント206、207、208のそれぞれのために、異なる読み書き動作を実行する3つのクライアント・スレッド216、217、218と、クライアント・スレッド216、217、218に対して、通知動作を実行する2つの通知スレッド220、222とを備える。概括して言えば、各クライアント・スレッド216、217、218は、通信回線230を通じて前記クライアントと通信するために、サーバ・コンポーネント・オブジェクト226、227、228をそれぞれ使用する。特定のクライアント・スレッド内のサーバ・コンポーネント・オブジェクトは、前記クライアントによってアクセスされている異なるコンポーネントの各々のための共有キャッシュ214内の異なるライトウェイトと通信し、このような各ライトウェイトに関連付けられたコンポーネント・データ・ラッパー (component data wrapper) を使用してこの通信を実行する。各コンポーネント・データ・ラッパーは、関連付けられたライトウェイトへのアクセスを制御し、1つ以上のクライアント・スレッドによってアクセスされ得る。

【0080】図7のシステムにおいては、クライアント206は、データ・ストア210、212内のコンポーネントを予約し、クライアント207は、データ・ストア210内のコ

ンポーネントを予約し、クライアント208は、データ・ストア212内のコンポーネントを予約している。図7に示すように、クライアント206のためのクライアント・スレッド216は、データ・ストア210と通信するライトウェイト234に順に接続された第1のコンポーネント・データ・ラッパー232と通信するために、サーバ・コンポーネント・オブジェクト226のうちの1つを使用する。確立の際、ライトウェイト234は、データ・ストア210内の前記データをコピーし、このデータを、該データを予約している各クライアントがアクセスできるようにする。クライアント・スレッド216のサーバ・コンポーネント・オブジェクト226のうちの1つは、データ・ストア212と通信するライトウェイト242に順に接続するコンポーネント・データ・ラッパー240と通信する。同様に、確立の際、ライトウェイト242は、前記データをデータ・ストア212からコピーし、このデータを、該データを予約している各クライアントがアクセスできるようにする。

【0081】クライアント207のためのクライアント・スレッド217は、コンポーネント・データ・ラッパー232と、従って、ライトウェイト234内の前記データ（即ち、データ・ストア210内の前記データ）にアクセスするために、ライトウェイト234と通信するためにサーバ・コンポーネント・オブジェクト227を使用する。一方、クライアント208のためのクライアント・スレッド218は、データ・ストア212に関連付けられたライトウェイト242内のデータにアクセスするためにコンポーネント・データ・ラッパー240と通信すべく、サーバ・コンポーネント228を使用する。このように、このコンフィギュレーションにおいて、クライアント206、207は、同一のコンフィギュレーション・コンポーネントを予約し、その結果、クライアント206、207の両方が、このコンポーネントのためにライトウェイト234にアクセスする。同様に、クライアント206、208は、同一のコンフィギュレーション・コンポーネントを予約し、両者が、このコンポーネントのためにライトウェイト242にアクセスする。その結果、ライトウェイト234、242は、2つのクライアント・スレッドによってそれぞれ共有され、異なるクライアントによって共有キャッシュ214内のデータの共有を可能にする。

【0082】更に詳細に説明するように、クライアントがデータベース203内のコンフィギュレーション・コンポーネントを予約するときに、サーバ202は、クライアント・スレッドを生成又は拡張するために、サーバ・コンポーネント・オブジェクト、コンポーネント・データ・ラッパー・オブジェクト、およびライトウェイトを確立する。生成の際、クライアント・スレッド216、217、218は、サーバ・コンポーネント・オブジェクト226、227、228からの読み出しおよびサーバ・コンポーネント・オブジェクト226、227、228への書き込みをなすべく、つまり、そこに生成された前記コンポーネント・データ

・ラッパーとライトウェイトとを介して共有キャッシュ214へ読み込み、および共有キャッシュ214から読み出すべく、前記クライアントによって使用される。共有キャッシュ214は、必要があれば、コンフィギュレーション・データベース203からの読み出しおよびコンフィギュレーション・データベース203への書き込みを行う。異なるクライアント・スレッドが各クライアントのために生成されるのであれば、各クライアント・スレッドは、クライアントのためにコンポーネント・データの一又は複数の予約を扱うことが可能である。通常、異なるサーバ・コンポーネント・オブジェクトは、クライアントによってコンポーネントの各予約のために生成される。

【0083】通知スレッド220は、データベース203になされた変更を検出し、そして、特に、データベース203内で保たれる変更リスト244の状態をチェックすることによってライトウェイトが生成しているデータ・ストア210、212になされた変更を検出する。変更がなされたとき、通知スレッド220は、変更が発生した変更コンポーネントを予約する前記クライアント・スレッドの各々へメッセージを送信させる。このような各クライアント・スレッドは、それから、適当なサーバ・コンポーネント・オブジェクトを使用して、そのクライアント・スレッドをクライアントに通知することが可能である。クライアントに送信された通知メッセージは、新しいコンポーネント・データを含むことが可能であるか、又は、単に新しいデータが存在すること、また、変更されたコンフィギュレーション・コンポーネントのために確立されたクライアント・スレッドを介してこのデータをクライアントが引くか又は読むかをしなければならないことを、クライアントに通知することが可能である。同様に、ランタイム・スレッド222は、如何なるランタイム・サービス246によって実行されるプロセス制御システムへなされる特定の変更をも検出し、該変更のデータに関連付けられたコンポーネントを予約しているクライアント・スレッドを通知する。

【0084】概括して言えば、コンフィギュレーション・データベース203内のデータへのアクセスは、コンポーネントに対してコンポーネント単位で実行される。各コンポーネントは、エリア、コントローラ、デバイス、制御モジュール等のようなコンフィギュレーション・データベース内の単一の項目であることが可能であり、望むならば、複数の項目の単一のツリーであることが可能である。さらに、任意のコンポーネントが親コンポーネントであることが可能であり、該親コンポーネントは、これに関連付けられた一又は複数の子コンポーネントを有すること、および／又は任意のコンフィギュレーション・コンポーネントが親コンポーネントの子コンポーネントであることが可能である。このように、エリアは、特定のエリアを示す多数の子コンポーネントを伴った親コンポーネントであることが可能である。各特定のエリ

アは、特定のエリアの中のコントローラ、ユーザ・インタフェース、制御モジュール等を示す子コンポーネントを伴った親コンポーネントであることが可能である。

【0085】さて、図8には、図7の共有キャッシュ通信システムを実現すべく使用されるオブジェクトのためのオブジェクト・マップが図示されている。特に、クライアント300（任意のクライアントであることが可能）は、例えば、図2のようなコンフィギュレーション階層でユーザに表示させる如き、クライアントによる任意の目的でアクセス又は予約されるコンフィギュレーション・データの任意の部分であるクライアント・コンポーネント・オブジェクト301を有することが可能である。子供側にてドットを有し、また、親側にてドットを欠如しているクライアント・コンポーネント301とフィードバック・パスによって示すように、クライアント・コンポーネント301は、それに関連付けられた一又は複数の子コンポーネントを有する親オブジェクトであることが可能である。図8において、ドットの欠如が1対1関係を示す一方、ドットの使用は1対1又は1対多の関係を示している。このように、クライアント・コンポーネント・オブジェクト301を図示するように、各子クライアント・コンポーネントが1つの直接の親コンポーネントだけを持つことができる一方、各親クライアント・コンポーネントは一又は複数の子クライアント・コンポーネントを持つことが可能である。

【0086】各クライアント・コンポーネント301は、クライアント300でのそのクライアント・コンポーネントに関連付けられたスマート・プロキシ302に結合されている。該スマート・プロキシ302は、クライアント・コンポーネント301に代わり、通信回線304を通じた通信を担っている。特に、前記スマート・プロキシは、コンポーネントに関連するクライアント要求を受ける分散オブジェクトであって、それらの要求を通信するサーバ306への通信回線304を通じてサーバ306を呼び出し、更新イベントを受信し、サーバ306から送信されるキャッシュ・データとメッセージとを格納し、このデータをクライアント・コンポーネント・オブジェクト301に渡す。勿論、スマート・プロキシ302は、任意の好ましい方法によって、又は任意の好ましい通信技術を使用する（「遅延認証通信および警報管理」というタイトルの米国特許出願に記述された遅延認証ルーチンを使用する）ことによって、通信回線304を通じて通信するために必要な操作を実行する。通信回線304は、遅い又は小さい帯域幅の回線（例えば、衛星、セルラー、インターネット、又はその他の共有広域ネットワーク回線）を含む任意の好ましい通信回線であることが可能であり、また、望ならば、間欠的な接続（例えば、モデム回線（接続））であり得るか、直接接続又は高速接続（例えば、専用のイーサネット回線）であり得る。

【0087】サーバ306は、通信回線304を通じたクライ

アントのスマート・プロキシ302との通信を担うサーバ・コンポーネント・オブジェクト308を含んでいる。各サーバ・コンポーネント308は、コンフィギュレーション・データベース内のコンフィギュレーション・コンポーネントの特性にアクセスするインタフェースを提供するために、また、クライアントに対して、イベントの更新を受け、これらのイベントをスマート・プロキシ302に転送するために、特定のクライアントによってアクセス又は予約される少なくともコンフィギュレーション階層の一部を記録又は保持することを担う。図8に示すように、サーバ・コンポーネント308は、親子関係を通じてその他のサーバ・コンポーネントに関連付けられることが可能である。そして、この親子関係は、関連付けられたクライアント・コンポーネント301のために確立された親子関係を反映する。特に、この関係は、クライアント階層（即ち、クライアントにより予約されているコンフィギュレーション階層）を反映する。各サーバ・コンポーネント308は、サーバ306の共有キャッシュ部分内でセットアップされたオブジェクトである唯一のコンポーネント・データ・ラッパー310に、着目され、また、通信可能に結合され得る。しかし、図8によって示されるように、コンポーネント・データ・ラッパー310は、異なるクライアント又はクライアント・スレッドのための多数のサーバ・コンポーネントに接続されることが可能である。コンポーネント・データ・ラッパー310は、共有データ（即ち、複数のクライアント又はクライアント・スレッドによって、図7の共有キャッシュ214内で共有されるデータ）へのインタフェースである。コンポーネント・データ・ラッパー310は、階層関係（全ての予約中のクライアントによって見られている全てのクライアント階層の上位集合（superset）である）を記録することが可能であり、サーバ・コンポーネントへのリファレンスを登録することが可能であり、変更イベントが検出されたときにサーバ・コンポーネントに通知することが可能であり、最後に修正された日付（コンフィギュレーション・データベースからのイベントを受け入れるか又は拒絶するために使用される）を記録することが可能であり、また、共有データをロックすることが可能である。サーバ・コンポーネント308と同様に、サーバ・コンポーネント308の全てとクライアント・コンポーネント301の全てとのために確立された関係を反映する関係をもって、コンポーネント・データ・ラッパー310は、親子関係において、その他のコンポーネント・データ・ラッパーに関連付けられる又は関連することが可能である。特定のコンフィギュレーション・コンポーネントにアクセスしている全てのクライアント・スレッドが同一のコンポーネント・データ・ラッパー310を通過するので、各コンポーネント・データ・ラッパー310は、多数のサーバ・コンポーネント308に接続され得ると言える。第1のクライアントがコンフィギュレーション・



データベース内で最初に項目を予約するときに、コンポーネント・データ・ラッパー・ファクトリ (component data wrapper factory) 312によって生成されるコンポーネント・データ・ラッパー310は、どのサーバ・コンポーネント308がコンフィギュレーション・データベース内の特定のコンポーネントを予約しているかをトラッキングし、予約中のサーバ・コンポーネント308の各々にこのデータへの変更について通知するように動作する。

【0088】各コンポーネント・データ・ラッパー310は、一又は複数のサーバ・コンポーネント308が予約しているコンフィギュレーション・データベース内で、コンポーネント・データのコピーを格納する唯一のライトウェイト・オブジェクト314に接続されるか、又は該ライトウェイト・オブジェクト314を含む。言い換えると、コンポーネント・データ・ラッパー310は、複数のサーバ・コンポーネント308の各々によるライトウェイト314のアクセスを取り扱うために、一又は複数のサーバ・コンポーネント308と、単一のライトウェイト314との間に挿入される。概括して言えば、ライトウェイト314は、サーバ・コンポーネント308間で供給されるデータのための貯蔵所であって、図7のコンフィギュレーション・データベース203になされた変更に関連する更新通知のために、図7の通知スレッド220とともに登録することに使用される。

【0089】コンポーネント・データ・ラッパー・ファクトリ312は、コンポーネント・データ・ラッパー310とライトウェイト314との取得／生成を担う。クライアントがそれがコンフィギュレーション・データベース203内でコンポーネントを予約したいことを示すとき、コンポーネント・データ・ラッパー・ファクトリ312は、このコンポーネントのためのコンポーネント・データ・ラッパー310が既に生成されているか否かを確認すべく最初にチェックする。もしそうならば、コンポーネント・データ・ラッパー・ファクトリ312は、このコンフィギュレーション・コンポーネントのためにクライアント・スレッドをコンポーネント・データ・ラッパー310に接続する。もしそうでなければ、コンポーネント・データ・ラッパー・ファクトリ312は、コンフィギュレーション・データベース203からの要求情報をライトウェイト314へロードさせるように、新しいコンポーネント・データ・ラッパー310とライトウェイト314とを生成し、この新しいコンポーネント・データ・ラッパー310と、関連付けられたライトウェイト314とを使用するクライアントのためのスレッドを生成又は拡張する。コンポーネント・データ・ラッパー・ファクトリ312が新しいラッパー310を生成するときに、コンポーネント・データ・ラッパー・ファクトリ312は、ラッパー310にユニークな不変の識別子を割り当てる。この識別子は、リファレンスしているサーバ・コンポーネント308と、これらの対応

するクライアント・コンポーネント301とへ渡されるものであって、グループの又は個々のコンポーネントをユニークに識別し、また、これに代えて、多くの煩雑な名前変更の問題を回避し、遅い通信回線 (例えば、ツリーの全てのツリー又はブランチ (枝) がクライアントで要求される時) を通じてIDのリストのより効率的な整列を提供する。

【0090】ロック・マネージャ316は、各コンポーネント・データ・ラッパー310に対して、異なるクライアントによってその関連付けられたライトウェイト314へのアクセスを制御させることを可能にする。ライトウェイト314をロックすることで、一又は複数の他のライトウェイトをロックすることを引き起こすことが可能であるので、例えば、親コンポーネント・データを格納するライトウェイトをロックすることで、その親コンポーネントのために子コンポーネントを保持するライトウェイトをロックすることを自動的に引き起こすことが可能である。同様に、各ライトウェイト314は、別のライトウェイトをロックする際に自動的にロックされることが可能であるので、その子コンポーネントのために親コンポーネントを保持するライトウェイトがロックされるときに、子コンポーネントを保持するライトウェイトがロックされることが可能である。このように、そのライトウェイトからの読み出し又は該ライトウェイトへの書き込みが任意の特定のクライアント・スレッド内で実行されるときに、ロック・マネージャ316によって、コンポーネント・データ・ラッパー310がそれに関連付けられたライトウェイト314をロックさせることが可能であり、このライトウェイトをロックすることで、他のライトウェイト (例えば、ロックされているライトウェイトに関連付けられた子コンポーネントを保持するライトウェイト) をロックすることを自動的に引き起こすことが可能である。

【0091】さらに、各ライトウェイト314は、コンテキスト・メモリ (context memory) を使用する図7のコンフィギュレーション・データベース203からの又は該コンフィギュレーション・データベース203へのコンポーネント特性の読み書きのための、一又は複数のトランザクショナル・メモリ・オブジェクト (XM: transactional memory object) 318を使用する。ライトウェイト314がコンフィギュレーション・データベース203に対して読み書きする必要がある都度、XM 318が始動される。該XM 318は、この読み書きをなすために使用するポインタを、前記コンテキスト・メモリ (ローカルのクライアント・スレッドである) に提供し、前記コンテキスト・メモリを使用するコンフィギュレーション・データベース203への読み込み又は書き出しをさせる。また、前記XM 318は、データベース203内のデータベース・オブジェクトを生成／削除するために、そして、コンフィギュレーション・データベース203に関係を持たせた通知を登録

解除するために使用され得る。適当な読み書きがライトウェイト314に一旦実行されると、XM 318は開放される。しかし、前記コンテキスト・メモリに格納されたデータは、未だ存在し、必要ならば、このメモリへのポインタは、後で使用されるスレッド・ローカル記憶装置に格納される。XMの使用については、オブジェクト・データベース・マネージャにおいては標準であって、ここでは更なる説明はしない。

【0092】通知エンジン320は、各XM 318に登録し、XM 318によって示されるコンフィギュレーション・データベース203内のデータ記憶装置の位置におけるデータ値の何れかへの変更を検出する。このような変更は、例えば、クライアントがコンフィギュレーション・データベース203への読み込みを引き起こすこと（ランタイム・プロセス又はその他の何らかの外部の処理等）によって、なされることが可能である。通知エンジン320は、これが、XM 318の何れかによって示されたデータベースの項目への変更を検出したときに、このXM 318と関連付けられたライトウェイト314を有するコンポーネント・データ・ラッパー310に通知し、変更がなされ、そして、コンポーネントのためのその新しい値が利用可能なコンポーネントを予約しているクライアントの全てに通知する。この操作を実行するために、通知エンジン320は、コンポーネント・データ・レジスタ（「ラッパー・マップ」とも呼ばれる）322を使用する。これは、XM 318によって示されたデータベース位置に、各コンポーネント・データ・ラッパー310をマッピングするマップである。より詳しくは、コンポーネント・データ・レジスタ322は、データベース203にアクセスしているXM 318の操作の結果として生成された通知クッキー又はIDにコンポーネント・データ・ラッパー310を関係付けるルックアップ・テーブルである。時々、これらのIDは、「マジック・ナンバー」とも呼ばれ、概括して言えば、これらのIDは、データベース203内に格納されたコンポーネントの位置又はそのものについて識別する。通知エンジン320は、マジック・ナンバーに関連付けられたデータベース位置の各々に関係する更新イベントのリストを得て、変更が発生した各コンポーネントに関連付けられたコンポーネント・データ・ラッパー310を見つけるためにコンポーネント・データ・レジスタ322を使用する。その結果として、通知エンジン320は、コンフィギュレーション・データベース203から定期的に変更リストを得て、変更が発生したコンポーネント・データ・ラッパー310への変更イベントを始動するイベント・ジェネレータである。

【0093】概括して言えば、クライアント・スレッドは、それによって変更通知を得るために生成されるので、各コンポーネント・データ・ラッパー310は、コンポーネント・データ・レジスタ322で、そのライトウェイトに関連付けられたXMの全てを登録する。代わりに、

コンポーネント・データ・ラッパー310は、ユニークなクッキーを登録毎に受け、これらのクッキーは、コンポーネント・データ・レジスタ又はラッパー・マップ322に格納される。変更通知に応じて、コンポーネント・データ・ラッパー310は、その関連付けられ得たライトウェイト314に対して、前記データベースから再ロードすることを、また、必要ならば、コンフィギュレーション階層を更新することを強制する。さらに、コンポーネント・データ・ラッパー310は、該コンポーネント・データ・ラッパー310に関係がある又はアクセスしているサーバ・コンポーネント308の各々へ更新メッセージを送信する。そして、サーバ・コンポーネント308は、それから、共有キャッシュ（即ち、ライトウェイト314）と再整合させる。最後に、サーバ・コンポーネント308は、クライアント・コンポーネント301に更新メッセージを送り、サーバ・コンポーネント308と再整合させる。

【0094】さて、図9を参照して、データベース・サーバ306（例えば、図7のデータベース・サーバ202、又は図1、図3～図6のデータベース・サーバの何れかであることが可能である）の状態は、例えば、任意のクライアントがコンフィギュレーション・データベース325（例えば、図7のデータベース203又は図1および図3～図6のコンフィギュレーション・データベースの何れかであることが可能である）内のデータの何れかを予約する前の、プレスタート（起動）時を示している。サーバ306は、その中にサーバ・コンポーネント・オブジェクトを有していないプレスタート・クライアント・スレッド326を含んでいる。つまり、共有キャッシュ327は、コンポーネント・データ・ラッパー又はライトウェイトおよび空のデータベース・コンテキスト・メモリ部分328を有していない。サーバ306の共有キャッシュ327は、データベース325から読まれているコンポーネントをロックすることを調整するコンポーネント・データ・ラッパー・ファクトリ312とロック・マネージャ316とを備えている。図9に示すように、通知スレッドは、上述した如きデータベース325内のデータになされた変更を検出する通知エンジン320、およびコンポーネント・データ・レジスタ、又はコンフィギュレーション・データベース325内の位置で共有キャッシュ327内のコンポーネント・データ・ラッパーをマッピングするラッパー・マップ322を備えている。勿論、データ・ラッパーは、共有キャッシュ327内に存在しないので、ラッパー・マップ322は、空である。

【0095】同様に、ランタイム・スレッドは、任意のサービス（例えば、自動検出サービス）であり得るランタイム・サービス338によって、又はランタイム動作若しくはコンフィギュレーション動作の間に実行するその他のアプリケーションによって、コンフィギュレーション・データの或る部分になされた変更を検出する通知エ

エンジン336と、ランタイム通知エンジン336に、サーバ306で実行しているクライアント・スレッド内の適当なコンポーネント・データ・ラッパーへ変更メッセージを送らせることを可能にするマップ340とを備える。起動時に、ランタイム通知エンジン336は、プロセス制御ネットワークのコンフィギュレーション内の特定の變更（例えば、新しいノードの追加又はシステム内の新しいデバイス（非稼働中のコントローラのような）の追加）を検出することを実行するオブジェクトの一例を生成する。1つの実施形態において、通知エンジン336は、非稼働中のコントローラと「Fieldbus」デバイスとのリストのためのランタイム・サービス338（それは、任意のランタイム・サービス又はアプリケーションであり得る）をポーリングする。マップ340は、一般に、適当なデータ・ラッパー（生成の際）がプロセス制御システム内の新しいノード、デバイス等の生成に関連する情報を受けるために登録する記憶位置又は記憶領域を指し示している。勿論、全てのデータ・ラッパーがランタイム・スレッドに登録する必要があるというわけではないが、新しく加えられたデバイス（例えば、図2に図示された非稼働中のコントローラのコンポーネント）に関係するコンポーネントに関連付けられたそれらのラッパーだけは登録する必要がある。起動時に、マップ340は、どのラッパーも登録されていない記憶位置を指し示している。

【0096】さて、図10においては、第1のクライアント342がコンフィギュレーション・データベース325内のコンポーネントに接続し、予約するときの、サーバ306の操作が図示されている。特に、クライアント・アプリケーションが、起動し、コンフィギュレーション階層内のコンポーネント348を要求するためにスマート・プロキシ346を使用している通信回線344を介してサーバ306に接続する。この際に、サーバ306は、プレスタート・クライアント・スレッド326をクライアント342に割り当て、サーバ306に接続する次のクライアントによって使用される新しいプレスタート・クライアント・スレッド（図示せず）を生成する。一般的に、クライアント342は、ルート・オブジェクト（例えば、エリア・コンポーネント）を要求することから始める。この時点で、エリア・サーバ・コンポーネント350は、サーバ306において生成され、エリア・サーバ・コンポーネント350は、ラッパー・ファクトリ312からエリア・コンポーネントを要求する。ラッパー・ファクトリ312は、コンポーネント・データ・ラッパー（以下、「ラッパー」と称す）が共有キャッシュ327内のエリア・コンポーネントのために生成されず、このようなラッパー354（エリア・ラッパー）を生成することを認識する。そして、ラッパー354をクライアント・スレッド326内のエリア・サーバ・コンポーネント350と結び付ける。ラッパー・ファクトリ312は、新しく生成されたラッパー354にユニークに関連付けられ、前記クライアント・スレッドのためにデータ

ベース・「サイト」・コンテキスト362に結び付けられたライトウェイト360を生成する。前記ルート・エリア・コンポーネントが要求されたので、前記サイト・コンテキストが使用される。そして、このコンポーネントは、データベース325内のサイト・オブジェクトに関連付けられる。この際に、ライトウェイト360は、XM（図示せず）を生成することによって前記エリア・コンポーネントのためにデータベース325に呼び出しを行う。前記XMは、クライアント・スレッド326によって使用されるべきコンテキスト・メモリ328内の記憶位置を指し示しており、前記データベース・マネージャに、前記エリア・コンポーネントを読み出させ、前記コンポーネント・データをサイト・コンテキスト・メモリ362に格納させる。次に、このデータが、ライトウェイト360にロードされ、前記XMが、開放される。

【0097】この処理の間、變更がそのデータベースの対応するもの（この場合、データベース325のエリア・コンポーネント内のもの）に発生するとき、ラッパー・ファクトリ312又は前記XMは、變更通知のためにラッパー・マップ322で新しく生成されたエリア・ラッパー354を登録する。この登録は、線363で図示されている。この登録を実行するために、ユニークな識別子（ID）がラッパー／データベース・サイトの組合せのために生成され、このIDが、通知スレッドに送信される登録メッセージ内に掲示される。登録メッセージの処理に際して、前記通知スレッドは、前記IDによってロックされたエリア・ラッパー354のために、エントリをラッパー・マップ322に加える。このエントリは、ラッパー354とともに、XM（データベース325内のオブジェクト又は位置を指し示す）に関連付けられたマジック・ナンバーに関係している。

【0098】次に、前記エリア・コンポーネントは、ラッパー354から、エリア・コンポーネント・データで更新されるサーバ・コンポーネント350へ送信される。サーバ・コンポーネント350は、それから、コンポーネント・データがクライアント・コンポーネント348に提供されるクライアント342のスマート・プロキシ346へエリア・コンポーネント・データを送信する。

【0099】その後で、前記通知スレッドは、データベース325内のエリア・コンポーネントへの変更を監視し、そして、變更がなされれば、ラッパー・マップ322によって定義されているように、データベース325内のこのコンポーネントに関連付けられたラッパーのためにIDを参照する。前記通知スレッドは、それから、變更について前記ラッパー（この場合、ラッパー354）に通知する。必要ならば、ラッパー354は、データベース325の読み出しを引き起こすことによって變更を反映するためにライトウェイト360を更新する。その後で、ラッパー354は、變更についてサーバ・コンポーネント350に指示し、サーバ・コンポーネント350は、前記エリア・コン

ポーネントのために新しいデータを要求することができる、そして、新しいデータは、ライトウェイト360から読み出され得る。次に、サーバ・コンポーネント350は、変更についてクライアント342に通知し、該クライアント342は、エリア・コンポーネントのための新しいデータをサーバ・コンポーネント350から要求するので、前記エリア・コンポーネントにおける変更は、自動的に、又は変更についてクライアントへ通知し、クライアントが新しいコンポーネント・データを要求するのを待機することによって、クライアント・コンポーネント348に反映される。クライアント342は、サーバ・コンポーネント350から新しいコンポーネント・データの手動で引き出すことができるが、変更が検出されたときに、新しいデータ又は変更されたデータがクライアントに自動的に送信されることについては、議論のために以下のように想定する。

【0100】同様に、前記ランタイム通知スレッドは、ランタイム・サービス338の操作に基づくプロセス制御システムのコンフィギュレーションにおいてなされた特定の変更を検出する通知エンジン336を使用する。特定の変更が発生した（例えば、ノード又はデバイスが加えられた）ならば、通知エンジン336は、検出された変更の表示を何れかのラッパーが受けるために登録したかどうか見るために、マップ340によってその変更のために指定された記憶位置を参照する。もしそうならば、通知エンジン336は、通知エンジン320によって生成されたイベント・メッセージが処理される方法と同様に、そのメッセージを処理することができる、登録したラッパーに、イベント・メッセージを送信する。ランタイム通知スレッドによって検出された変更の通知を受けたいデータ・ラッパーが、マップ340によって、即ち、例えば、データ・ラッパーが生成されたときに、その記憶位置での変更通知を受けるために前記ラッパーのIDを格納することによって、その変更のために指定された記憶位置で登録すると言える。

【0101】さて、図11において、クライアント342は、次に、エリア・クライアント・コンポーネント348のために子供をロードすることを要求し得る。この場合、クライアント342は、子供コマンドおよびエリア・ルーツ・コンポーネントのロードを呼び出し、このコマンドは、プロキシ346を介してサーバ・コンポーネント350へ渡され、そこから、エリア・ライトウェイト360へ渡される。エリア・ライトウェイト360は、前記エリア・子コンポーネントについての情報を得るためにクライアント・スレッド326に対するコンテキスト・メモリ362を使用するコンフィギュレーション・データベース325へ問い合わせる。このデータは、返され、そして、ライトウェイトと関連付けられたデータ・ラッパーとは、それから、エリア・コンポーネントに関連付けられた各子コンポーネントのために生成される。図11において、

前記エリア・コンポーネントは、「Area\_A」と名付けられた「Fred」という説明付きの1つの子コンポーネントだけを含むように示されている。そして、この新しい子コンポーネント・データは、新しいライトウェイト370に格納される。しかし、その他のエリア・子コンポーネントは、また、エリア・コンポーネントに関連付けられることができ、そして、この場合には、新しいライトウェイトは、下記のその他のオブジェクトとともに、これらの子供の各々のために生成される。

【0102】新しいライトウェイト370の作成の際、エリア・ラッパー354（又はラッパー・ファクトリ312）は、子ラッパーとしてエリア・ラッパー354に関連付けられた別々のラッパー372内の新しいエリア・ライトウェイト370をラップ（wrap）する。また、エリア・ラッパー354は、サーバ・コンポーネント350に前記新しいエリア・コンポーネントを返す。新しいライトウェイト370は、エリア・子コンポーネントに関連する特定のデータを得るためにコンテキスト・メモリ376を介してデータベース325で登録すべくXMを生成する。また、ラッパー372又はライトウェイト370は、新しいラッパーID（この場合、1つのラッパーID）と、これに関連付けられたデータベース・マジック・ナンバー（この場合、1つの新しいマジック・ナンバー）とのリストとともに、登録メッセージを前記通知スレッドに送信する。前記新しいIDは、線378によって示するように、新しいラッパー372を示すために前記ラッパー・マップに格納される。また、データベース325における新しい子エリア・コンポーネントの位置を示すマジック・ナンバーは、ラッパー・マップ322に提供され、このIDのために格納される。

【0103】一方、エリア・サーバ・コンポーネント350は、新しい子エリア・コンポーネントのための子エリア・サーバ・コンポーネント380を生成し、そうする際に、子サーバ・コンポーネント380のために生成されたラッパー372へ、新しい子エリア・サーバ・コンポーネント380内でポインタを埋め込む。また、サーバ・コンポーネント350は、前記通信回線とプロキシ346とを介してクライアント・エリア・コンポーネント348へ子コンポーネント（この場合、1つ）のリストを送信する。そして、エリア・クライアント・コンポーネント348は、子エリア・クライアント・コンポーネントを生成し、新しい子クライアント・コンポーネント382と新しい子サーバ・コンポーネント380との間での通信を可能にするために子エリア・クライアント・コンポーネント382において、サーバ・コンポーネント・リファレンスを（サーバ・コンポーネント380に）埋め込む。クライアント・コンポーネント382のためのスマート・プロキシ384は、それから、生成され、データベース325内で新しい子コンポーネントのための予約を含むためにクライアント・スレッド326を拡張する処理を完了する。勿論、同じ手順は、特定のコンポーネントのその他の子コンポー

ネットのために発生する。そして、複数の子供が存在するならば、クライアント・スレッド326は、複数の子コンポーネントを予約するために拡張され得る。同様に、クライアント・スレッド326は、新しく予約された子コンポーネント382に関連付けられた子コンポーネントを得ることによって、又は別のルーツコンポーネントを要求することによって、更に拡張され得る。

【0104】さて、図12を参照して、第2のクライアント386が、第1のクライアント342により予約された同一のコンポーネントを予約する際の処理について、詳述する。ここでは、サーバ306の通知スレッドは、明確化のために図示していない。特に、第2のクライアント386は、第1のクライアントと同様の方法でサーバ306に接続し、前記エリア・コンポーネントを予約する。サーバ306への接続に際しては、第2のクライアント386は、第2のプレスタート・スレッド387に割り当てられる。この例では、クライアント386は、第2のクライアント・スレッドでエリア・サーバ・コンポーネント388を生成するエリア・ルーツ・コンポーネントを求める。サーバ・コンポーネント388は、エリア・ルーツ・コンポーネントのためのラッパーをラッパー・ファクトリ312に求める。なぜなら、ラッパー354は、前記エリア・ルーツ・コンポーネントのための共有キャッシュ327内で既に存在するからであり、ラッパー・ファクトリ312は、ラッパー354にポインタを返す。この時点で、エリア・サーバ・コンポーネント388は、第1のクライアント342のために既に確立されたライトウェイト360から読み出されるエリア・ルーツ・コンポーネント・データについてラッパー354に問い合わせる。ライトウェイト360に格納されたエリア・ルーツ・コンポーネント・データは、エリア・ルーツ・サーバ・コンポーネント388に返され、そして、そこから、クライアント・エリア・コンポーネント389に返される。

【0105】エリア・ルーツ・コンポーネント389の子供のための第2のクライアント386による要求に際して、エリア・ルーツ・サーバ・コンポーネント388は、ラッパー・ファクトリ312に子ラッパーを要求する。そして、これらのラッパー（この場合、1つのラッパー）が既に存在するので、ラッパー372へのポインタは、エリア・ルーツ・サーバ・コンポーネント388にラッパー・ファクトリ312によって提供される。それから、サーバ・コンポーネント388は、ラッパー372へのポインタを有する子エリア・サーバ・コンポーネント390を生成する。エリア・ルーツ・サーバ・コンポーネント388は、適当なクライアント・コンポーネント392を生成する第2のクライアント386とサーバ・コンポーネント390と通信するスマート・プロキシ393とに子供の情報を送信する。また、子エリア・サーバ・コンポーネント390は、ラッパー372を介してライトウェイト370を読み出し、プロキシ393を介してクライアント・コンポーネント392に

このデータを送信する。この処理においてはデータベース325への呼び出しがなされなかったので、第2のクライアントのデータベース・コンテキストは空である。つまり、全ての情報が、そのデータを得るために第1のクライアント・スレッド326のデータベース・コンテキストを使用したライトウェイト360、370を介してやって来たのである。しかし、ライトウェイトが生成されなかったというデータを第2のクライアント386が要求するならば、ラッパー・ファクトリ312は、そのコンポーネントのためのラッパーおよびライトウェイトを生成し、データベース325からそのデータを得るために第2のクライアント・スレッド387のためのデータベース・コンテキストを介してデータベース325に接続する。

【0106】エリアルーツ又は子エリア・コンポーネントを現在予約するその他の任意のクライアントは、データベース325に如何なる呼び出しもすることなく、共有キャッシュ327からこのデータにアクセスすることができ、これは、データベース325上でのロードを低減すると言える。同様に、イベント通知を通知スレッドから受けるのに際して、データ・ラッパー354は、クライアントの両方ともに変更を通知するために、第2のクライアント386のためのサーバ・コンポーネント350に、また、第1のクライアント342のためのサーバ・コンポーネント388に通知する。

【0107】さて、図13を参照して、コンフィギュレーション・データ項目への変更を検出する通知スレッドに関連付けられた動作と、この変更通知が各々の予約中のクライアントへ自動的に返される方法とについて説明する。この場合、第2のクライアント386は、「Area\_A」の名前を「Fred」から「Wilma」に変更し、サーバ306へこの変更とともに、書き込み又は起用メッセージを提示する。サーバ・コンポーネント390は、新しい名前をロードし、ライトウェイト370において新しい名前をロードするラッパー372に、この名前を付与する。ライトウェイト370は、第2のクライアント386のためのデータベース・コンテキスト394に自動的に新しい名前をロードする。ここから、前記新しい名前は、データベース325に格納される。しかし、「Area\_A」コンポーネントの第1のクライアントのバージョンには、何も発生しなかった。過去に、第1のクライアント342は、変更のためにデータベース325に定期的にポーリングする必要があったので、データベース325からの遅い通信回線を通じた多くの読み出しと多くのダウンロードという結果となる。

【0108】しかし、この場合、通知エンジン320は、ラッパー・マップ322においてマジック・ナンバーに関連付けられたデータベースの項目の各々のために、更新イベントのリストのためにデータベース325に定期的にポーリングする。各イベントは、どのオブジェクトが変更されたかについて示す情報と、変更の種類（名前の変更、削除、特性の変更、子供の追加／削除等）のような

前記変更についての幾つかの情報と、多分、名前の変更操作の場合の新しいおよび古い名前のような幾つかの手掛かり (hint) とを含んでいる。通知エンジン320は、ラッパーIDとともに変更されたオブジェクトのマジック・ナンバーに一致するラッパー・マップ322を使用し、前記関連付けられたラッパー (この場合、ラッパー372) に更新イベントを処理するように求める。この処理は、このオブジェクトの新しい値又は変更された値を得るためにデータベース325からライトウェイト370を再ロードすることを含むことが可能である。このような再ロードは、ライトウェイト370がデータベース325への変更に関与していなかったときに必要となる。ラッパー372は、例えば、検出された変更が古いか若しくは期限切れの変更であるか、又はその他の任意の好ましい理由であるならば、ライトウェイト370を再ロードすることを拒否することが可能である。また、ラッパー372は、前記変更についてサーバ・コンポーネント380、390の各々に通知するか、これに代えて、メッセージ (例えば、ウィンドウ・メッセージ) を (このコンポーネントを予約する) クライアント342、386の各々へ送信する。ラッパー372からイベント・メッセージを処理するときに、サーバ・コンポーネント380、390は、イベントおよび任意の新しいデータを、対応するクライアント342、386のスマート・プロキシへ転送する。第1のクライアント342のプロキシ384は、第1のクライアント342におけるユーザ・インタフェース上での再表示を引き起こさせるクライアント・コンポーネント382に再表示メッセージを掲示する。この場合、ユーザ・インタフェースが再表示するときに、プロキシ384が新しいデータをクライアント・コンポーネント382に譲り渡し、更新が完了する。第2のクライアント386は、前記変更を既に持つので、前記メッセージを無視することが可能である。

【0109】この予約システムで、第1のクライアント342は、如何なるデータベースの呼び出しもする必要もなく、第1のクライアント342での前記変更された項目の完全な再表示を早めるために、唯1回の呼び出しが (例えば、遅い通信回線を介して) サーバ306から第1のクライアント342へなされただけである。勿論、この処理は、全てのライトウェイトについて、また、任意のライトウェイトを予約している全てのクライアントについて発生する。従って、データベース325への如何なる変更をも、最小のデータベースの読み出しと、遅い通信回線を介した最小の通信とを伴って、その項目を予約する各クライアントへ自動的に送信することができる。

【0110】さて、図14を参照して、キャッシュ327への並列アクセスの仲介について、ロック・マネージャ316の操作を更に詳細に記述する。前述の説明によってよく理解されるように、共有キャッシュ327におけるライトウェイトの各々は、一又は複数のクライアント・スレッドによって共有され得る。その結果、多くのクライ

アントが、同一のライトウェイトから同時に読み出しを行うことができる。ロックについては、これらの読み出しには一般に必要ではない。しかし、2つのクライアントが、ライトウェイトに、つまりデータベース325に同時に書き込みを行おうとする状況に出くわすことがあり得、これは、デッドロック (deadlock) のような問題を引き起こす。

【0111】これらの問題を防ぐために、ロック・マネージャ316は、ロックを渡して、ロックテーブルを保守することによって、前記ライトウェイトの各々へのアクセスを管理する。概括して言えば、ロック・マネージャ316は、ラッパー上でのロック要求を与えることによって動作する。クライアント・スレッドがライトウェイトにアクセスしたいときには、前記スレッドは、前記ラッパー上でのロックの読み書きを最初に取得しなければならない。例えば、別のクライアント・スレッドがそのライトウェイトにアクセスするプロセスにおいて存在するために、前記ロックが利用できないならば、ロック・マネージャ316は、前記ロックが自由になるまで追い詰められて、要求しているスレッドを保持する。

【0112】図14の例においては、第1および第2のクライアント342、386は、「Area\_A」の説明を書き込もうとしている。ここで、第2のクライアント386は、ライトウェイト370への書き込みを実行しようとし、この処理においては、ロック・マネージャ316からのラッパー372上でロックを要求し、これを取得する。第1のクライアント342が、それから、ライトウェイト370の特性からの読み出し又は該特性への書き込みをしようとするならば、ロック・マネージャ316は、このスレッドがデータ・ラッパー372上での読み出しロック又は書き込みロックを取得しようとしたときに、第1のクライアント・スレッド326の実行を待機する。一方、第2のクライアント・スレッド387は、「Area\_A」の名前を「Wilma」に変更するために、それ自身のデータベース・コンテキスト394を介してライトウェイト370へ書き込む。この書き込み操作が完了したときに、ロック・マネージャ316は、第1のクライアント・スレッド326に対して、ライトウェイト370の、つまり、データベース325の読み書きを進めさせる。読み出しロックが開放されるまで、書き込みロックは拒否又は保留され得ることに注意する。しかし、読み出しロックが別のクライアント・スレッドに与えられているならば、読み出し要求を保留する理由は通常ない。また、ロック・マネージャ316が上述した予約操作を確立するために使用され得る点に注意する。特に、予約ロックは、例えば、ライトウェイト370によって要求され、また、ライトウェイト370上のクライアントへ与えられることが可能である。これ以後においては、任意のクライアントが、ライトウェイト370の読み出しを行うことが可能であるが、該ライトウェイト370への書き込みはできない。予約ロックが与えられたならば、ロッ

ク・マネージャ316は、ラッパー372に対して、書き込みロックを要求しているクライアントへメッセージを送信するようにさせ、異なるクライアントによって予約中であるように、このデータへの書き込みは許容されない。勿論、ロック・マネージャ316は、また、前記予約ロックを与えられたクライアントの識別を提供又は格納することが可能であり、また、起用操作の間、このクライアントによるライトウェイト370への書き込みを可能とする。

【0113】ロックがあれば、何処であってもデッドロックの可能性がある。例えば、第1のクライアントが、コンポーネントAのロックを取り出し、同時に、第2のクライアントが、コンポーネントAのロックを取り出す。次に、第1のクライアントが、コンポーネントBのロックを取り出そうとするが、第2のクライアントによってコンポーネントBのロックが開放されるまで停止する。しかし、コンポーネントBのロックが開放される前に、第2のクライアントが、コンポーネントAのロックを取り出そうとするならば、両方のコンポーネントAおよびコンポーネントBのロックは開放されることがなく、従って、デッドロックとなる。共有キャッシュ327におけるデッドロックを回避するために、ロック・マネージャ316には、如何なる特定のクライアントであっても、ロックを順々に取り出し、開放しなければならないような制約を持たせることが可能である。しかし、この場合、2つのクライアントは、他方によってなされた変更を上書きすることが可能であり、この結果、後者の書き込みが勝つ。例えば、第1のクライアントが、コンポーネントAを修正し、第2のクライアントがコンポーネントBを修正する。第1のクライアントが、それから、コンポーネントBを修正し、第2のクライアントがコンポーネントAを修正する。デッドロックが回避される一方、コンポーネントAは、第2のクライアントの変更を有し、コンポーネントBは、第1のクライアントの変更を有する。

【0114】ライトウェイトは階層を持たないが、幾つかのライトウェイト間リファレンスが存在することが可能である。もしそうならば、この事実は、例えば、更新操作が同時にロックすべき両方のライトウェイトを要求するならば、デッドロックの潜在的な要因である。この問題を回避するために、ロック・マネージャ316は、関連したライトウェイトに関してロック注文を実行することができる。例えば、第1のライトウェイトが第2のライトウェイトに関連付けられるならば、第1のライトウェイトは、第2のライトウェイトのロック・リストに加えられる。クライアントが第2のライトウェイトをロックしたいならば、前記クライアントは、最初に第1のライトウェイトをロックしなければならない。ロック・マネージャ316は、この注文のリストを保持し、ロック手順の注文を実施することが可能である。この従属ロックを実施するために、関与したオブジェクトが従属ツリーを

形成しない限り、ロック・マネージャ316は、クライアントにつき1つのロックのみを一度に許容することが可能である。

【0115】データベース325において一連のコンポーネントを修正することは、1つのデータベース処理について発生する。それは、処理が失敗する可能性があり、従って、変更がデータベース325においてなされる前に、データベースの状態をロールバック (rollback) し、キャッシュされたコンポーネント (例えば、ライトウェイト) がデータベース325に格納されたものと矛盾する結果となる。ロールバック・メカニズムを提供するために、クライアント・スレッドは、データベース修正処理が稼動している間、前記コンポーネントを加えた、修正されたコンポーネントのリストを保持することが可能である。複数ステップのデータベース処理が中止されるか、又は失敗と見なされたならば、前記リストが横渡しされ、各コンポーネントが再ロードされる。これは、コンポーネントの特性を再ロードし、新しい/削除された子供についてチェックすることである。このような修正されたコンポーネントのリストが既知の順序 (例えば、ラッパーIDに応じて) に保たれるならば、ロックテーブルを生成し、クライアント操作につき複数のロックを引き出すことが可能である。

【0116】ライトウェイトがロックされる時間を減らすために、ライトウェイトは、データベース325へ特性を書き戻すとき、ライトウェイトに格納されたコンポーネント・データ・オブジェクトのクローンを最初に生成することが可能である。該クローンは、新しい特性の値を供給され、該値をデータベース325に格納するために使用される。この操作が一旦完了したならば、第1のコンポーネント・データ・オブジェクトは、変化されるか、又は前記クローンを優先して差し替えられる。前記クローンが作られた後で、前記ライトウェイトをロック解除することができるので、このようなクローン技術を使用することは、キャッシュ327又は前記ライトウェイトがロックされる時間を減じ、前記キャッシュへのアクセスの並列性を改善する。しかし、この状況には、2つのクライアントが、同時に、同一のライトウェイトへの書き込みをしようとする点で、競争条件が存在することになる。特に、第1のクライアントが、第2のクライアントの前に、データベース325へのコンポーネントのそのバージョンを保存するが、第2のクライアントの後で、前記キャッシュ (即ち、ライトウェイト) を変化させるならば、前記キャッシュは、データベースで矛盾することになる。ここで、前記データベースは、第2のクライアントの変更を有するが、前記ライトウェイトは、第1のクライアントの変更を有する。この状況は、ロックを調整することによって、又はいつでも任意のライトウェイトの1つのクローンだけを許容することによって、回避され得る。これらの手順の全ては、ロック・マネージャ

ャ316によって仲介され得る。

【0117】上述のように、新しい子ライトウェイトは、クライアント要求に応じて、又はクライアントが新しい子供を加える時に、データベース325からこれらの子供をロードすることによって生じる。望むならば、クライアントは、親をロックする前に、新しい子コンポーネント・データ・オブジェクトを生成／ロードする時点で、キャッシュ327を操作することを許容し得る。従って、前記親ライトウェイトが、子供が前記階層に実際に接続されているときか、又は前記親に関連付けられたときにだけ、ロックされる必要がある。言い換えると、全ての子供について完了するまで、生成されている子コンポーネントに関連付けられたオブジェクトの全てが、非共有方式において又はローカル・メモリにおいて生成されることが可能である。それから、前記親は、これらの子供へのポインタを含み、これによって、これらの子オブジェクトを前記共有メモリへ移すために、ロックおよび修正されることが可能である。その結果、新しい子コンポーネントに関連付けられるために変えられるとき、且つ子オブジェクトの作成の間でないときにのみ、前記親ライトウェイトはロックされる。

【0118】さて、図15には、ランタイム操作によってなされたプロセス制御システムのコンフィギュレーションへの変更について、クライアントに通知するために動作するランタイム通知スレッドの操作を示している。この場合、クライアント400が非稼働中のデバイス・コンポーネントを予約し、クライアント・コンポーネント402、プロキシ404、前記非稼働中のデバイス・サーバ・コンポーネント406、ラッパー408、およびライトウェイト410が、クライアント400およびサーバ306内に既にセットアップされているものと仮定する。また、ラッパー408が生成されたとき、該ラッパー408が、新しいコントローラのシステムへの追加の指示を受けるためにマップ340に登録されているものと仮定する。この例では、前記ランタイム通知スレッドの通知エンジン336が、ランタイム・サービス（例えば、自動検出アプリケーション）を介して加えられた新しいコントローラ（「CTRL001」と名付けられている）の追加を検出し、この新しいコントローラが存在を示す非稼働中のデバイス・ラッパー408に更新イベントを送信する。非稼働中のデバイス・ラッパー408は、前記検出されたコントローラのために子ラッパー412を生成することによって、新しいコントローラのためにライトウェイト414を生成することによって、そして、もし可能であれば、ライトウェイト414にランタイム・サービスによって届けられるようなコントローラ・データを格納することによって、前記更新イベントに回答する。また、ライトウェイト414は、データベース325のこのデータを格納することが可能であり、望むならば、このデータを、データ・ラッパー412に、および新しいコントローラ情報が前記通知スレッド

のためにラッパー・マップ322を格納するデータベース位置のマジック・ナンバーに、登録することが可能である。また、ラッパー412は、マップ340に更新を登録することが可能である。共有キャッシュ327に新しい子供としての新しいコントローラをロードすることに際して、イベントが、新しい子データ・ラッパー412にポインタを埋め込み、新しいコントローラ・データをクライアント400へ押す、新しい子コントローラ・サーバ・コンポーネント416を生成することによって応答する、前記非稼働中のデバイス・サーバ・コンポーネント406に提示される。クライアント・プロキシ404は、新しい子イベントを受けて、新しい子供の情報を格納し、クライアント400でのユーザ・インタフェースに再表示メッセージを提示する。非稼働中のデバイス・クライアント・コンポーネント402が新しい子供をロードするときに、該非稼働中のデバイス・クライアント・コンポーネント402は、新しい子コントローラ・コンポーネント418を生成し、非稼働中のデバイス・サーバ・コンポーネント406によってクライアント400に送信されたサーバ・コンポーネントの識別情報を使用するコントローラ・コンポーネント418のために、サーバ・コンポーネント416への接続を確立する。コントローラ・クライアント・コンポーネント418のプロキシ420は、生成され、また、コントローラ・クライアント・コンポーネント418に代わって、サーバ・コンポーネント416との通信を想定する。このように、通知エンジン336によって検出された前記コントローラの追加およびその変更は、ライトウェイトに格納されたデータを変更することが可能であり、また、ライトウェイトおよび関連付けられたラッパーの生成又は削除を引き起こす、関連付けられたデータ・ラッパー408に送信される。新しいラッパーおよびライトウェイトの作成が、親のオブジェクトを予約している各クライアント・スレッドを拡張する結果となる。

【0119】上述した操作の他に、その他の動作又は操作は、サーバ306によって実行されることが可能である。例えば、クライアントは、該クライアントがデータベース325に格納すべきコンポーネントを生成する生成操作を、実行することが可能である。この生成動作に関連付けられた操作は、データベース325からこのオブジェクトをロードする代わりに、データベース325の根底にあるデータベース・オブジェクトを生成する特別なステップとともに、子供をロードすることと非常に似ている。ここで、前記クライアントは、コンポーネントを生成し、生成したコンポーネントの親に関連付けられたサーバ・コンポーネントに、このコンポーネントを送信する。この動作によって、ラッパー・ファクトリ312が、前記関連付けられた親のラッパーとライトウェイトとの子供として、ラッパーとライトウェイトとを生成し、新しいライトウェイトが、新しいコンポーネントをロードする。前記ライトウェイトは、それから、通知スレッド



におけるラッパー・マップ322でのデータベース位置とラッパーとを登録するのと同じように、新しいコンポーネントをデータベース325に書き込む。新しいサーバ・コンポーネントは、それから、ラッパーのために生成され、この新しいコンポーネントのためにクライアントとの通信を仮定する。

【0120】データベース325からコンポーネントを削除するために、クライアントは、関連付けられたサーバ・コンポーネントを最初に削除し、それから、開放する。サーバ・コンポーネントの削除は、削除された状態で、前記関連付けられたデータ・ラッパーとライトウェイトとを置き、それは、それから、データベース325からコンポーネントを削除する。前記ラッパーは、その他の参照しているサーバ・コンポーネント（その他のクライアント・スレッド内の）の各々に削除メッセージを送り、予約中のクライアントに前記削除動作を通知する。前記削除上のとき、前記ラッパーは、コンフィギュレーション階層から除去され、更なる任意のデータベース操作も実行することができない。しかし、全ての参照しているサーバ・コンポーネントがそれを開放するまで、前記ラッパーは、残ることが可能である。勿論、前記参照しているサーバ・コンポーネントは、前記サーバ・コンポーネントが前記削除動作のクライアントに通知した後で、発生する、前記関連付けられたクライアントの開放まで、開放されない。勿論、データベース・イベント・メカニズムは、前記通知スレッドの通知エンジンによってラッパーに1箇所に集められる削除メッセージを最終的に生産することが可能である。前記ラッパーは、削除された状態にあるので、それはこのメッセージを無視する。

【0121】コンフィギュレーション・コンポーネントの幾つかの操作によってそのコンポーネントの種類が変化するので、該コンポーネントの変化を生じさせることが可能である。この変化は、前記コンポーネントとサブ・ツリーとをデータベース325から再ロードすることを必要とする。この操作を生じさせるために、前記変化されたコンポーネントのための全てのライトウェイトの親/子階層は、既存のライトウェイトへデータベース325から再ロードされる。その後で、前記サーバ・コンポーネントは、再確立され、これらのサーバ・コンポーネント内の新しい値は、それから、前記クライアント・コンポーネントと整合される。このような整合は、子コンポーネントを通じて繰り返すこと、その種類が変更され、もはや存在しないものを除去することを含む。新しい子供は、ロードされて、必要に応じて加えられる。

【0122】勿論、クライアントは、単に開放され、又はデータベース325内のコンポーネントを予約解除することが可能である。この場合、前記クライアントは、前記関連付けられたサーバ・コンポーネントを開放し、そ

れによって、前記関連付けられたラッパーがラッパーの参照回数を減少させるようになる。前記ラッパーのためのラッパーのリファレンス（参照）回数が零になると、サーバ・コンポーネントは、このラッパーを参照せず、従って、このラッパーを予約しない。この場合、前記ラッパーはアンロード（unload）する。望むならば、前記ラッパーは、該ラッパーが直ちに再ロードされる必要がある場合に備えて、アンロードする前にタイムアウト時間を待つことが可能である。アンロードするとき、前記ラッパーは、その関連付けられたライトウェイトを破壊する（データベース325内の更新通知の登録を取り消し、前記ラッパー・マップからの登録を取り消す）。さらに、コンポーネントが開放されるとき、それは、そのライトウェイトと全ての子ライトウェイト/子ラッパーを開放し、子ライトウェイト/子ラッパーのバック・ポインタを零にする。

【0123】ここでは、サーバ306の操作が、1つ又は2つのコンポーネントにアクセスする2つのクライアントのために記述されている一方、同様の技術が、拡張されることができて、データベース内の任意の数のコンフィギュレーション・オブジェクトから読み出し、また、該任意の数のコンフィギュレーション・オブジェクトへの書き込みを行う任意の数のクライアントおよびクライアント・スレッドに使用され得ると言える。さらに、複数のクライアントの予約者関係を確立するために共有キャッシュを使用する技術は、プロセス制御システム内でコンフィギュレーション・データにアクセスするものとしてここに記述されている一方、この技術は、コンフィギュレーション・データベースと関係がないもの、また、プロセス制御データベースと関係がないものを含むその他のどの様なデータベース・アクセス・システムにおいても使用されることが可能であると言える。

【0124】また、ここに記述されているデータベース・アクセス技術は、クライアントがコンポーネント単位でサーバと通信すると仮定している一方、コンフィギュレーション・ツリーの整列は、クライアントとサーバとの間の、より最適な通信を提供するために使用されることが可能である。特に、クライアントがツリー又はサブ・ツリーの全体を引き出さなければならないということを知っているときには、1つのサーバがツリー全体を得るために呼び出しをすることは、むしろ、ツリーの各レベルで子ルーチンのロードの呼び出しを再帰的に行うことよりも効率的である。クライアントがツリー全体をサーバに書き込みたいときには、前記ツリー全体が信号メッセージで送信されることが可能であるように、逆もまた真実である。勿論、これらのメッセージはクライアントとサーバとによって横取りされることが可能である。そして、予約関係を確立するために必要なオブジェクト（先に述べた）は、前記ツリー又はサブ・ツリーの全体を予約するために、また、ツリー又はサブ・ツリーの全

体に書き込むために、前記クライアント又は該クライアントに必要なサーバ内のオブジェクトの全てを直ぐに生成することが可能である。

【0125】また、前記サーバおよびクライアントによって作成又は実行されるべく、ここに記述されているオブジェクト、その他の要素、およびステップは、任意の好ましいプログラミング技術又はプログラミング言語を使用して実行されることが可能であると言える。そして、これらのプログラムは、メモリに格納され、ここに記述されている前記クライアント、サーバ、およびデータベース内において任意の方法でプロセッサに実行されることが可能であると言える。さらに、ここに記述されているデータ・アクセス技術は、ソフトウェアにおいて実現することが好ましいが、ハードウェア、ファームウェア等により実現されられても、また、プロセス制御システム10に関連付けられた任意のプロセッサにより実現してもよい。このように、望むならば、この技術は、標準的な多目的CPU又は特別に設計されたハードウェア又はファームウェアにより実現することも可能である。ソフトウェアにより実現するときには、前記ソフトウェア・ルーチンは、磁気ディスク、レーザー・ディスク、光ディスク、又はその他の記録媒体、更には、コンピュータ又はプロセッサのRAM、ROM等のような任意のコンピュータが読み取り可能なメモリに格納されることが可能である。同様に、例えば、コンピュータが読み取り可能なディスク若しくはその他の可搬型のコンピュータの記憶メカニズムを介したものの、又は電話線、インターネット等のような通信チャネルを通じたもの（可搬型の記録媒体を介してこのようなソフトウェアを提供することと同一視又は置き換え可能である）を含む任意の公知の又は好ましいデータの移動方法を利用して、このソフトウェアをユーザ又はプロセス制御システムに届けることも可能である。

【0126】このように、本願発明は、説明する目的から、特定の例に関して記述されているものであって、本願発明を限定するものではない。また、本願発明においては、その目的および範囲から逸脱することなく、開示されている実施形態の変更、追加、又は削除がなされ得ることは、当業者にとっては明らかである。

【0127】

【発明の効果】本願発明に係るコンフィギュレーション・データベース・システム、分散型のコンフィギュレーション・データベース・システム、コンフィギュレーション・データを格納および使用する方法、データベース・サーバ、データベース・コンポーネントへのアクセスを提供する方法、データベースに格納されたコンポーネントへの並列アクセスを提供するためにデータベース・サーバを使用する方法、およびプロセス制御システムにおいては、以上に詳述したように、コンフィギュレーション・データベースが、プロセス制御システム内の複数

の物理的な位置に分散された複数のデータベースを備え、各データベースが、コンフィギュレーション・データのサブセットを格納する。該サブセットには、前記システム内の任意のサイトでのユーザからアクセスすることができる。共有キャッシュを有するデータベース・サーバは、前記データベースからの最小回数の読み込みのみで、複数の予約者にコンフィギュレーション・データを読み込ませるようにアクセスする。前記システム内で予約者によって見られているコンフィギュレーション・データが古くなることを防止すべく、前記データベース・サーバは、前記データベース内の各項目への変更を自動的に検出し、なされた変更の通知を予約者の各々に送信する。これにより、それがコンフィギュレーション・データベース内に実際に存在するかのように、ユーザがコンフィギュレーションの状態を常に見ることができ

る。

【図面の簡単な説明】

【図1】 衛星通信回線を介して通信可能に接続された2つの地理的に離隔したサイトを備えるプロセス制御システムのブロック図である。

【図2】 コンフィギュレーション・アプリケーションで示される典型的なプロセス制御システム階層の図である。

【図3】 遅い通信回線を介して接続され、共通のマスター・コンフィギュレーション・データベースを共有するメイン・サイトおよびリモート・サイトを備えるプロセス制御システムのブロック図である。

【図4】 共通のマスター・コンフィギュレーション・データベースを共有するメイン・サイトと2つのオフライン・サイトを備えるプロセス制御システムのブロック図である。

【図5】 共通のマスター・コンフィギュレーション・データベースを共有する、ローカル・エリア・ネットワークを介してプラント・サイトに接続され、間欠的な通信接続を介してユーザ・インタフェースに接続された制御室を備えるプロセス制御システムのブロック図である。

【図6】 地理的に離隔した異なる部分を遅い又は小さい帯域幅の通信回線を介して通信可能に接続されたプロセス制御システムにおいて使用される分散型のコンフィギュレーション・データベース・システムのブロック図である。

【図7】 複数のクライアントとコンフィギュレーション・コンポーネントとの間でコンフィギュレーション・データベース内に格納される予約者／公開者通信を行うコンフィギュレーション・データベース・サーバ内で、共有キャッシュを用いるコンフィギュレーション・データベースのクライアント／サーバ・システムのブロック図である。

【図8】 図7のコンフィギュレーション・データベー

ス内に格納されたコンフィギュレーション・コンポーネントへアクセスする複数のクライアントを提供するために、図7のクライアント／サーバ・システム内で用いられるオブジェクト・モデルを示す図である。

【図9】 プレスタート（起動）時における図7のデータベース・サーバ内のオブジェクトの状態を示すブロック図である。

【図10】 第1のクライアントがコンフィギュレーション・データベースから第1のコンフィギュレーション・コンポーネントにアクセスすべくサーバに接続する際の、図7のデータベース・サーバ内のオブジェクトの操作を示すブロック図である。

【図11】 第1のクライアントがコンフィギュレーション・データベースからコンポーネントの子供をロードする際の、図7のデータベース・サーバ内のオブジェクトの操作を示すブロック図である。

【図12】 第2のクライアントが、既にデータベース・サーバの共有キャッシュに格納されたコンフィギュレーション・データベースから、第1および第2のコンフィギュレーション・コンポーネントにアクセスすべくデータベース・サーバに接続する際の、図7のデータベース・サーバ内のオブジェクトの操作を示すブロック図である。

【図13】 データベース・サーバの通知スレッドによって生成されたイベント通知に応じた、図7のデータベース・サーバ内のオブジェクトの操作を示すブロック図である。

【図14】 データベース・サーバ内で共有キャッシュへの並列アクセスの仲介をする図7のデータベース・サ

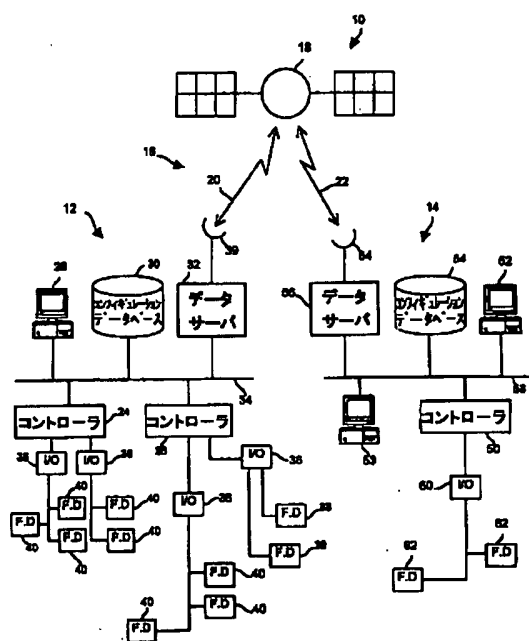
ーバ内のロック・マネージャの操作を示すブロック図である。

【図15】 ランタイム・サービスの通知スレッドによって生成されたイベント通知に応じた、図7のデータベース・サーバ内のオブジェクトの操作を示すブロック図である。

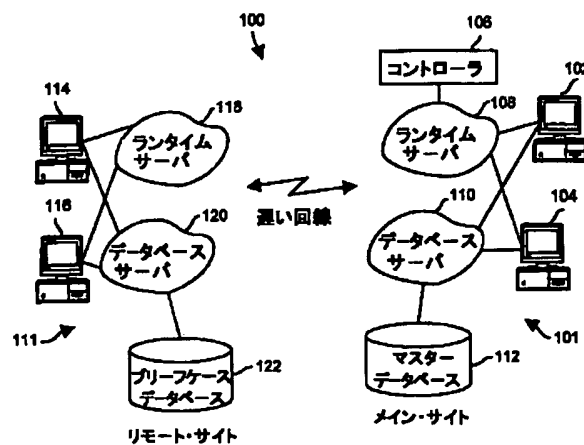
【符号の説明】

10, 130, 145, 170 プロセス制御システム  
12, 131 メイン・サイト（第1のサイト、ローカル・サイト）  
14, 135, 136 リモート・サイト（第2のプロセス制御サイト）  
16, 213, 230, 304, 344 通信回線  
24, 26, 50 プロセス・コントローラ  
30, 132, 147 マスター・データベース  
54, 142, 156, 162 ブリーフケース・データベース（ミラー・データベース）  
65 ツリー  
133, 138, 148, 152, 164, 202, 306 データベース・サーバ  
150, 154 ランタイム・サーバ  
210, 212 データ・ストア  
214 共有キャッシュ  
220 通知スレッド  
222 ランタイム・スレッド  
316 ロック・マネージャ  
320, 336 通知エンジン  
327 共有キャッシュ

【図1】

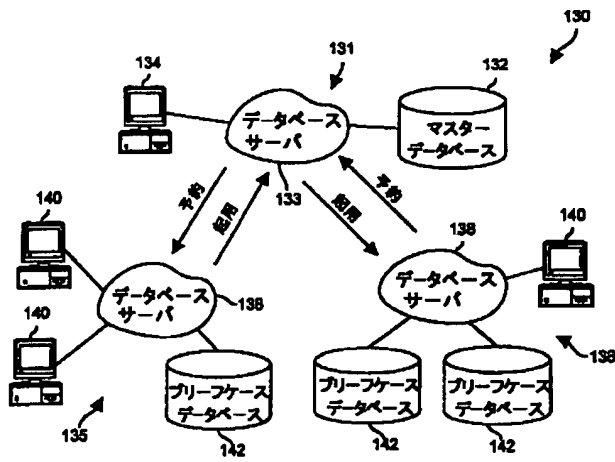


【図3】

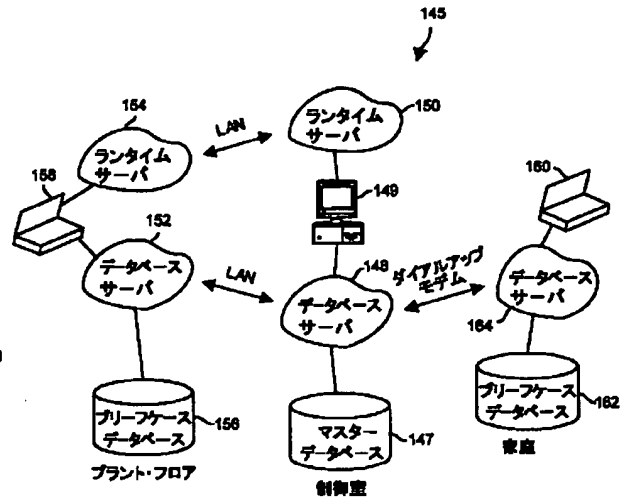




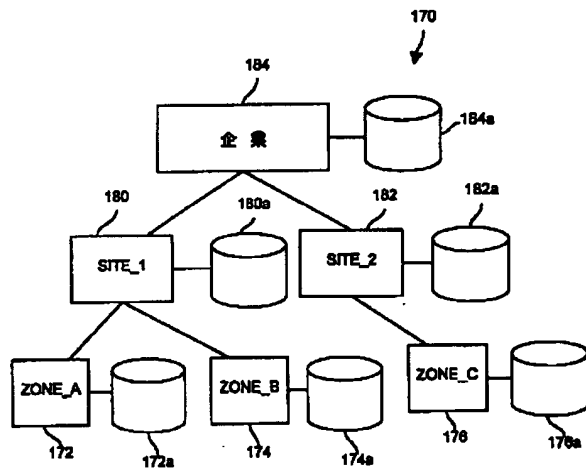
【図4】



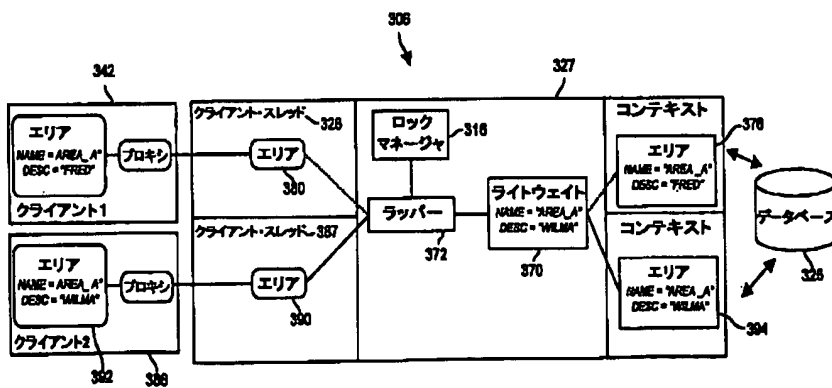
【図5】



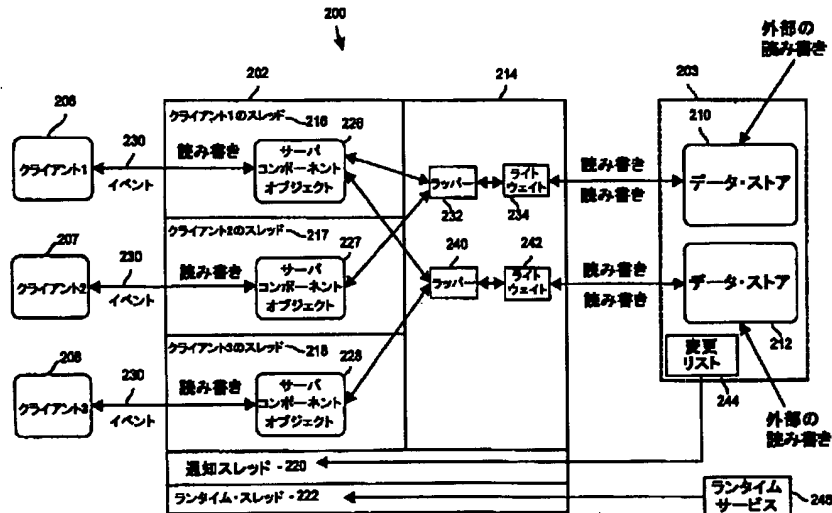
【図6】



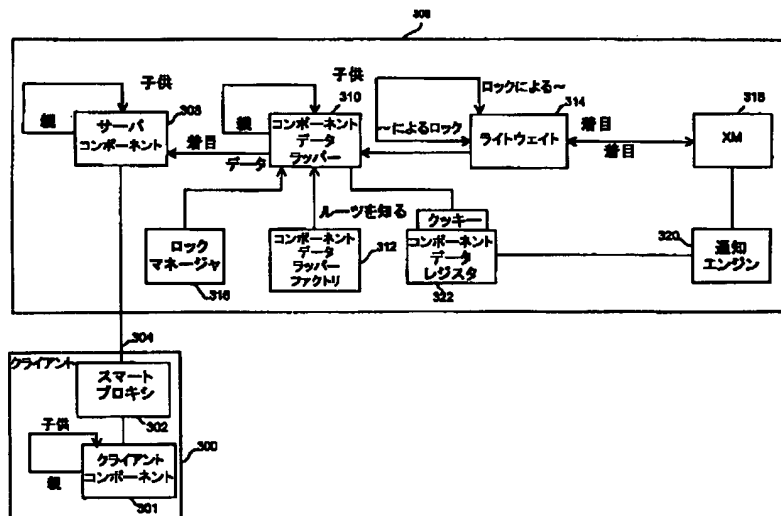
【図14】



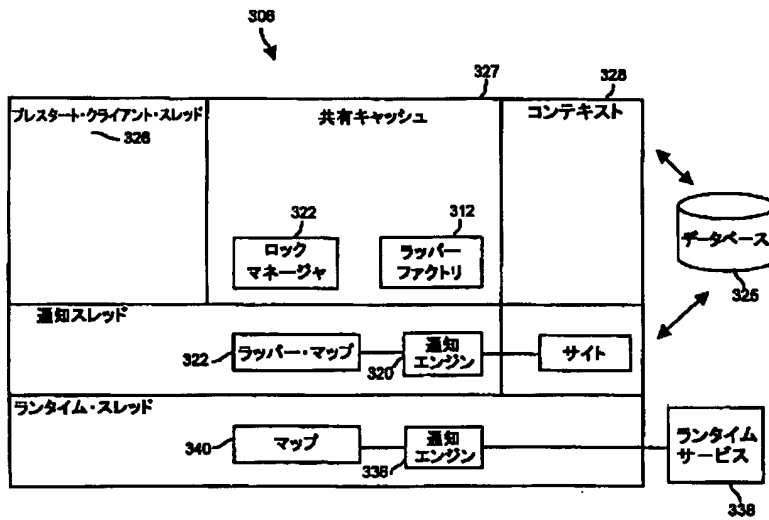
【図7】



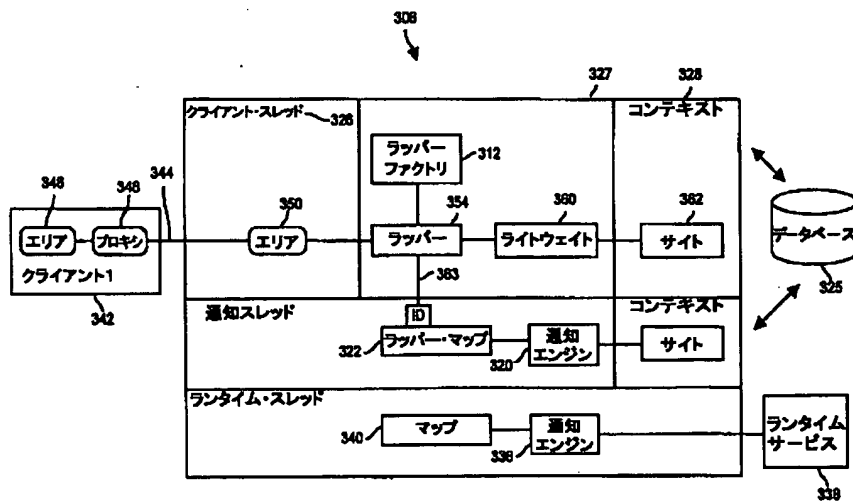
【図8】



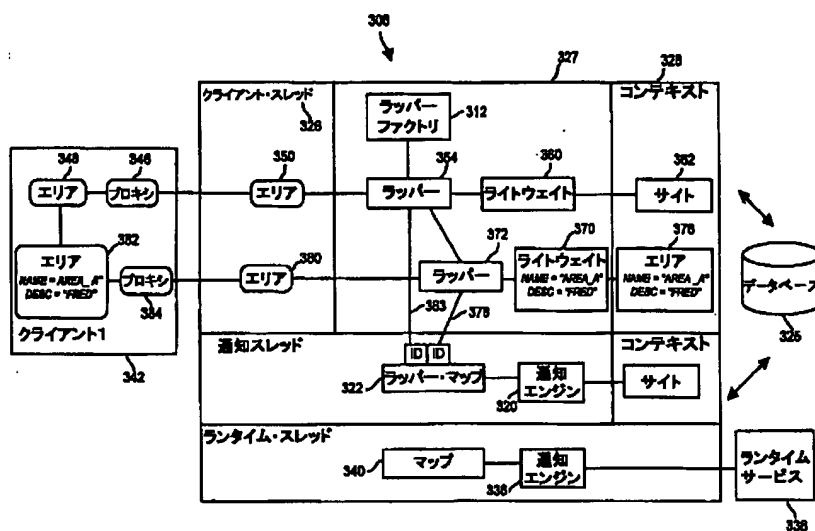
【図9】



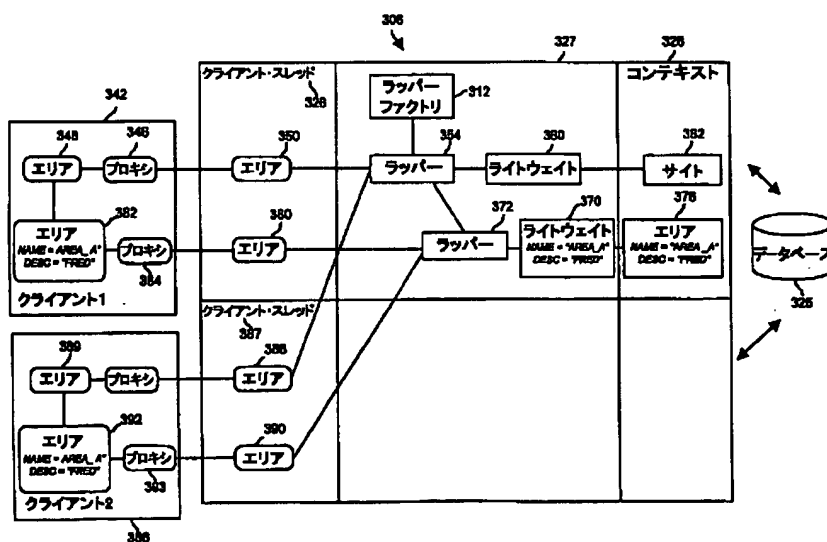
【図10】



【図11】

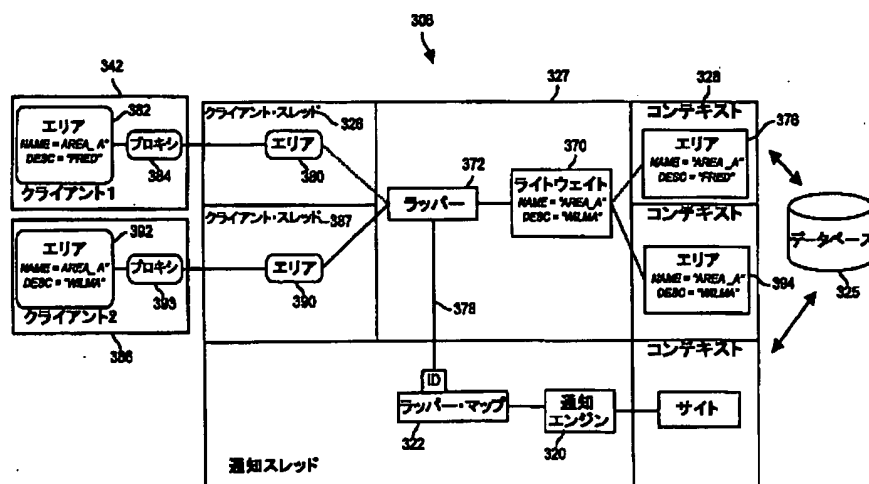


【図12】

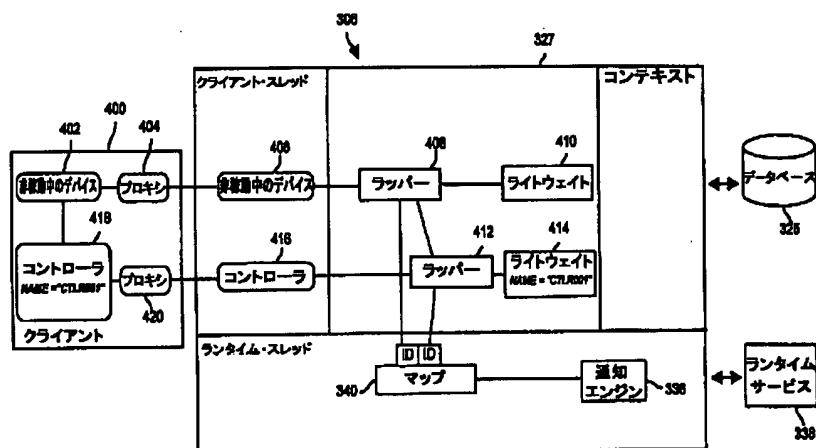




【図13】



【図15】



フロントページの続き

(72)発明者 チャットコフ, テレサ  
アメリカ合衆国 78759 テキサス オー  
スティン コードラ ドライブ 4002

(72)発明者 ギルバート, スティーヴン  
アメリカ合衆国 78758 テキサス オー  
スティン マウンテン クウェイル ロー  
ド 10109

【発明の名称】